Raport dla Klubu Rzymskiego:
Mikroelektronika w urządzeniach wojskowych

P 1877/83

7-8 1983

informatyka

Automatyczne redagowanie tekstów
LSI-11
Programowanie procesów współbieżnych
GOBI
Symulator ODRA-RIAD
CESARO
Historia oprogramowania

WYDAWNICTWO SIGMA

Płatne rubryki

Propozycję tę kierujemy do dużych instytucji informatycznych, liczących się na polskim rynku — instytucji usługowych, produkujących sprzęt, oprogramowanie, szkolących kadry itd., do wszystkich, których działalność ma szersze, nie tylko lokalne czy doraźne znaczenie.

Proponujemy im stałą obecność na łamach INFORMATYKI, pod własną winietą i — za pieniądze. W przypadku potentatów (ELWRO, ZETO, PTI, MERA-Błonie, ETOB, uczelnie...) — w każdym (lub prawie każdym) numerze, w innych — co któryś — ukazywałaby się firmowana przez daną instytucję jedna kolumna (strona w piśmie) lub jakaś jej część.

Umowa między nami określałaby liczbę kolumn w roku, jaka miałaby być dla danej instytucji przeznaczona. Opłata za jedną kolumnę wynosi 13,2 tys. zł.

Proponujemy, aby każda decydująca się na współpracę z nami instytucja wybrała przedstawiciela, który reprezentowałby w redakcji interesy swojej firmy. Byłby to więc rodzaj rzecznika prasowego i — jednocześnie — koordynatora wydawnictw. Zapewne bowiem dałoby się pogodzić pracę tej osoby przy przygotowaniu materiałów dla redakcji z innymi jej obowiązkami edytorskimi, a chyba także — z normalnymi, zawodowymi.

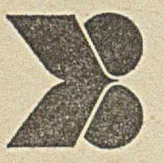
Dzięki takiej innowacji moglibyśmy liczyć na zwiększony przepływ informacji i poglądów. Instytucje same określałyby swój profil wydawniczy (oczywiście, z naszą pomocą), my zaś mielibyśmy gwarancję systematycznego napływu materiałów. I — co ważne — przestalibyśmy przynosić deficyt.

Prosimy o niezwłoczne rozpatrzenie naszej oferty. I szybki kontakt. Pomyślne ustalenia — jak sądzimy — przyniosą korzyści nie tylko partykularne.

REDAKCJA

NACZELNA ORGANIZACJA TECHNICZNA

WYDAWNICTWO



SIGMA

CZASOPISM I KSIĄŻEK TECHNICZNYCH

ul. Świętokrzyska 14a
00-950 Warszawa
skrytka pocztowa 1004

KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ

mgr inż. Zbigniew GLUZA, dr Janusz GWIAZDA, Władysław KLEPACZ (zastępca redaktora naczelnego), dr inż. Tomasz PAWLAK, mgr Marek SOBCZYK, mgr Andrzej SZAŁAS, mgr Zbigniew ŚWIRSKI, dr inż. Janusz ZALEWSKI

Sekretarz redakcji: mgr Teresa JABŁOŃSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BAŃKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. Antoni MAZURKIEWICZ, gen. dr inż. Marian PASTERNAK, dr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, prof. dr hab. Tadeusz WALCZAK

Materiałów nie zamówionych Redakcja nie zwraca.

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pok. 243 i 244, tel. 27-71-40, dyżury redakcji 10.00—12.00

Zakł. Graf. „Tamka". Zam. 2180. Obj. 7,0 ark. druk. Nakład 3300 egz. M-100

Cena egzemplarza zł 150.— INDEKS 36124 Prenumerata roczna zł 900.—

P. 1877/83

Informatyka

zastosowania w gospodarce, technice i nauce

Nr 7-8
MIESIĘCZNIK
1983
ROK XVIII
Lipiec-Sierpień

ORGAN KOMITETU INFORMATYKI, MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI ORAZ KOMITETU NAUKOWO-TECHNICZNEGO NOT DS. INFORMATYKI

W NUMERZE: Strona

Automatyczne redagowanie tekstów
Andrzej W. Abramowicz 2

Raport dla Klubu Rzymskiego
— Zastosowanie mikroelektroniki w urządzeniach wojskowych
Frank Barnaby 5

Programowanie procesów współbieżnych (1). Środowisko sprzętowe
Zygmunt Liszyński 8

Przegląd mikroprocesorów 16-bitowych (5). LSI-11
Jerzy Grabowski 11

Projektowanie systemu graficznego. Część 2
Jacek Owczarczyk 15

Standaryzacja języka BASIC (3). Rozszerzenia modułu rdzeniowego
Wacław Iszkowski, Marek Maniecki 19

SYSTEMY

GOBI — system do obliczeń geometrycznych
Andrzej Fąbijanek, Krzysztof Marciniak, Jerzy Wojciechowski 23

Wieloprogramowy dyskowy symulator ODRY 1305 na R-32
Marek Dziedzic, Krzysztof Perycz, Jerzy Wiliński 25

CESARO — system analizy i rozpoznawania obrazów wizualnych
Piotr Pachowicz, Ryszard Tadeusiewicz 27

Warunki i metody efektywnego zastosowania systemu TPP
Emilia Fladrowska 30

MC-82 makrogenerator o przeznaczeniu ogólnym
Witold Klaudel 33

HISTORIA

Wielkie wydarzenia w historii oprogramowania
Oprac. Jan Wierzbowski 34

ALGORYTMY

Generator zmiennej losowej o rozkładzie normalnym
Jadwiga Orylska, Antoni Miklewski 37

Generowanie par liczb z rozkładem normalnym
Jakub Tatarkiewicz 38

ZE ŚWIATA

PRINTEMPS INFORMATIQUE czyli informatyczna wiosna
Marianna i Marek Sobczykowie 38

USA, Europa Zachodnia, Japonia — rynek elektroniczny
Oprac. Jan Ryżko 41

Mikrosystemy
Oprac. Jacek Walas 45

IBM PERSONAL COMPUTER
Oprac. Marek Sobczyk 47

Informatyzacja rozliczeń finansowych
Grażyna Kuśmierz, Maciej Leśny 48

Mniej instrukcji — większa szybkość? (*Oprac. MkS*) 50

RECENZJE

Niedoceniana poufność
Marek Sobczyk 51

Projektowanie programów poprawnych i dobrze zbudowanych
Andrzej Szałas 51

TERMINOLOGIA

Słownictwo z zakresu inżynierii oprogramowania
Janusz Zalewski 52

W SKRÓCIE III okł.

ANDRZEJ W. ABRAMOWICZ
Instytut Maszyn Matematycznych
Warszawa

Automatyczne redagowanie tekstów

Rosnące koszty tradycyjnych metod produkcji poligraficznej i malejące koszty sprzętu informatycznego powodują, że automatyczne redagowanie tekstów jest jedną z najdynamiczniej rozwijających się dziedzin zastosowań informatyki. W chwili obecnej celowość wprowadzania techniki automatycznej realizacji wydawnictw, zarówno w aspekcie technicznym, jak i ekonomicznym, nie budzi żadnych wątpliwości, a gama istniejących na tym polu rozwiązań, w krajach wysoko rozwniętych, jest już bardzo bogata.

SYSTEMY AUTOMATYCZNEGO REDAGOWANIA TEKSTÓW

Ważnym etapem produkcji poligraficznej jest przygotowywanie form kopiowych. Obecnie uzyskuje się je najczęściej przez naświetlanie tekstu na odpowiednim materiale światłoczułym, a następnie wywoływanie naświetlonego materiału. Wśród urządzeń służących do naświetlania można wyróżnić:

- naświetlarki mechaniczno-optyczne, wyposażone w negatywy wzorców znaków, źródło światła i odpowiedni układ optyczny
- naświetlarki z liniowym zapisem uzyskiwanym za pomocą lampy elektropromieniowej
- naświetlarki laserowe, w których za pomocą promienia laserowego eksponowany jest jednocześnie cały wiersz tekstu.

W przygotowywaniu form kopiowych najbardziej długotrwałymi i pracochłonnymi procesami są składanie i łamanie tekstu. Głównie te właśnie procesy są automatyzowane za pomocą sprzętu informatycznego i metod przetwarzania danych tekstowych, przy czym stosowane są zarówno automaty, jak i uniwersalne maszyny cyfrowe wyposażone w odpowiednie środki programowe. Automaty są stosunkowo prostymi urządzeniami, ukierunkowanymi na konkretny typ wydawnictw. Podzielić je można na:

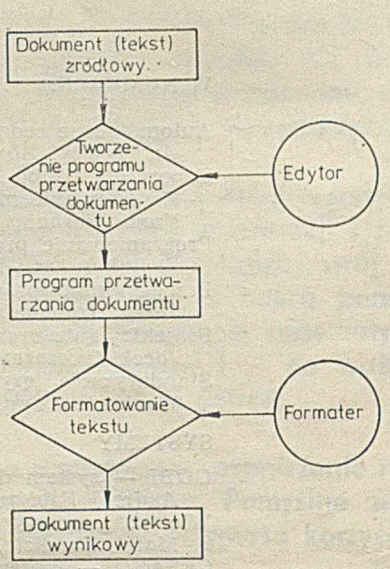
- automaty o ograniczonych możliwościach funkcjonalnych, będące wyposażonymi w procesor i pełniącymi funkcje redagowania naświetlarkami, dla których dane przygotowywane są na maszynowych nośnikach informacji za pomocą pracujących autonomicznie urządzeń kodujących
- automaty wielofunkcyjne, realizujące funkcje wprowadzania, korekty i redagowania tekstu, połączone bezpośrednio z urządzeniem wyjściowym (drukarką lub naświetlarką).

W dalszym ciągu zajmować się będziemy wyłącznie systemami komputerowymi realizowanymi przy użyciu maszyn cyfrowych ogólnego przeznaczenia.

Automatyczne składanie i łamanie tekstu jest cechą wspólną wszystkich systemów formatowania (redagowania). Proces automatycznego przetwarzania dokumentów tekstowych można przedstawić schematycznie w sposób następujący:

Można więc wyróżnić trzy fazy przetwarzania:

- tworzenie programu przetwarzania dokumentów
- automatyczne formatowanie
- wyprowadzanie sformatowanego tekstu.



Schemat procesu automatycznego przetwarzania tekstów (tekst ujęty w rombach powinien być umieszczony w prostokątach — przyp. A.A.)

W związku z tym środki programowe stosowane w systemach przetwarzania tekstów dzielą się na edytory i formatery.

Edytor jest narzędziem stosowanym przy tworzeniu i korekcie tzw. programu przetwarzania tekstu (dokumentu). Program ten składa się z danych wejściowych (dokument źródłowy) i wstawionych w odpowiednie miejsca rozkazów języka programowania automatycznego składu tekstu. Język programowania automatycznego składu tekstu służy do opisu algorytmów składania i łamania tekstu. W praktyce stosuje się zarówno edytory standardowe, jak i wyspecjalizowane, realizujące szeroki zakres funkcji pozwalających użytkownikowi na łatwe manipulowanie jednostkami tekstowymi, a w szczególności — dopisywanie, usuwanie, przemieszczanie, kopiowanie, zamianę miejsc oraz wyszukiwanie określonych fragmentów tekstu.

Charakterystycznymi przykładami rozbudowanych edytorów są EDIT/360, wchodzący w skład systemu IBM Text Processor, oraz TECO — implementowany na PDP-11.

Formater realizuje automatyczne kształtowanie form typograficznych. Najogólniej rzecz biorąc — wykonując instrukcje języka programowania automatycznego składu tekstu, dzieli on tekst dokumentu źródłowego na wiersze (justowanie) i strony (stronicowanie, paginacja). Edytor natomiast jest funkcjonalnie zupełnie nie związany z formaterem, a służy jedynie — jak już wspomniano — jako narzędzie do przygotowania programu składu. W szczegól-

Mgr inż. ANDRZEJ W. ABRAMOWICZ jest absolwentem Wydziału Elektroniki Politechniki Warszawskiej, ze specjalnością informatyka. Od ukończenia studiów (1975 r.) pracuje w Instytucie Maszyn Matematycznych w Warszawie. Zajmował się problemami oprogramowania telekomunikacyjnego, a obecnie pracuje nad zagadnieniami przetwarzania danych tekstowych.

ności więc programy mogą być przygotowywane na innych urządzeniach, oczywiście przy zachowaniu zgodności kodu, chociaż i tego można uniknąć, jeśli system dysponuje odpowiednim preprocesorem.

Wprawdzie często stosowane są systemy kompleksowe, zawierające wyspecjalizowany edytor i formater, niemniej obiektem zainteresowań teoretyków i programistów zajmujących się przetwarzaniem danych tekstowych są języki programowania automatycznego składu tekstu i formatery. Efektem działania formatera jest dokument zawierający rozkazy pewnego języka pośredniego, który w następnej fazie zostanie przetworzony przez postprocesor do postaci umożliwiającej sterowanie konkretnym, pracującym w trybie on- lub off-line, urządzeniem wyjściowym. W systemach przystosowanych do obsługi jednego urządzenia, lub do obsługi kilku urządzeń tego samego typu (o niewiele różniących się parametrach), postprocesor wkomponowany jest często w formater i tworzy z nim jedną całość.

W zależności od przeznaczenia, komputerowe systemy przetwarzania tekstów można podzielić na:
- jednostanowiskowe
- wielostanowiskowe z przetwarzaniem lokalnym
- wielostanowiskowe z przetwarzaniem zdalnym (wykorzystujące transmisję danych).

Wśród tych ostatnich stosowane są (głównie w USA) bardzo rozbudowane systemy tworzące sieci o różnej organizacji, łączące znacznie oddalone od siebie agencje, redakcje i drukarnie.

Według tego samego kryterium wyróżnić można systemy:
- ukierunkowane na konkretny typ wydawnictw
- uniwersalne, wyposażone w złożone formatery, pozwalające na konstruowanie różnorodnych skomplikowanych form typograficznych, dostosowane do współpracy z wieloma urządzeniami wyjściowymi.

Można stosować następujące urządzenia wyjściowe:
- drukarki, często o rozbudowanym repertuarze znaków i funkcji
- urządzenia znakowo-graficzne
- urządzenia fotoskładu (konwencjonalne i laserowe).

Pod względem trybu pracy systemy można podzielić na:
- interakcyjne
- wsadowe, przetwarzające uprzednio wprowadzony, kompletny program składu.

Tryb interakcyjny stosowany jest zarówno w prostych rozwiązaniach, pozwalających na bieżącą ingerencję operatora w sposób justowania wierszy, jak również w systemach dostosowanych do składu gazetowego, gdzie wybór tego trybu pracy jest podyktowany z jednej strony specyfiką samego składu, z drugiej zaś — częstą koniecznością aktualizowania informacji napływających ciągłym strumieniem. Systemy formatujące bez ingerencji operatora przetwarzają wprowadzony do systemu kompletny program składu i zatrzymują pracę jedynie w przypadku błędów semantycznych, uniemożliwiających dalsze przetwarzanie. Są to często systemy uniwersalne, nie związane bezpośrednio z konkretnym urządzeniem wyjściowym.

Przed zainicjowaniem procesu formatowania operator wprowadza niezbędne informacje, np.: nazwę urządzenia wejściowego i nazwę zbioru (zbiorów) źródłowego, wersję językową, nazwę urządzenia i zbioru wyjściowego, dyspozycję dotyczącą drukowania wykazu ewentualnych błędów, wersję formatera.

Systemy automatycznego przetwarzania danych tekstowych mogą być używane do realizacji praktycznie dowolnej klasy dokumentów, a więc opracowań naukowych, technicznych, ekonomicznych czy literackich. Obszarem ich zastosowań może być również formatowanie: listów, notatek (np. handlowych), wykazów, faktur, sprawozdań, komunikatów, rozpraw, dokumentacji, broszur, biuletynów, książek oraz czasopism.

Formatery systemów mogą być dostosowane do przetwarzania dokumentów zawartych w bazach danych, dają więc możliwość automatycznej generacji np. wykazów bibliograficznych, związanych z informacjami cytowanymi w tekście źródłowym; inne, dostosowane do formatowania listów czy notatek handlowych, mogą umieszczać automatycznie pewne specyficzne dla tego typu wydawnictw zwroty bądź nagłówki, bez potrzeby ich umieszczania w dokumencie źródłowym.

JĘZYKI PROGRAMOWANIA SKŁADU I FORMATERY TEKSTÓW

Niewątpliwe korzyści płynące z automatyzacji procesów wydawniczych spowodowały powstanie i dynamiczny rozwój nowej specyficznej dziedziny oprogramowania aplikacyjnego — przetwarzania danych tekstowych. Problemami tymi zajmują się obecnie zarówno ośrodki uniwersyteckie, jak i firmy produkujące sprzęt i oprogramowanie maszyn cyfrowych. Opracowanie dobrego języka automatycznego składu tekstu i — co się z tym wiąże — formatera jest, być może wbrew pozorom, bardzo trudnym przedsięwzięciem, szczególnie gdy formater ma być narzędziem uniwersalnym, pozwalającym na realizację skomplikowanych form typograficznych, a zarazem działającym niezależnie od urządzenia wyjścowego. Należy zdać sobie sprawę z tego, że podstawowym założeniem jest całkowita automatyzacja składania i łamania tekstu.

Już sam problem dzielenia tekstu na wiersze wiąże się z określonymi trudnościami. Zadaniem programu realizującego tę czynność jest wybór optymalnych punktów zakończenia wierszy z wykorzystaniem, gdy zajdzie potrzeba, automatycznego podziału wyrazów na sylaby. W dodatku, kryteria wyboru optymalnego punktu podziału zmieniają się w zależności od formatu i obiektów występujących w wierszu. W procesie formowania stron złożonych z wyjustowanych wierszy najwięcej trudności powoduje składanie i umieszczanie pagin żywych, skomplikowanych tabel, formuł matematycznych, wzorów (np. chemicznych), a przede wszystkim przypisów. Chociaż istnieją określone liczne reguły dotyczące składania i łamania tekstów, zawarte w branżowych normach poligraficznych, to jednak tego rodzaju zadania są bardzo trudne do zalgorytmizowania.

Najwcześniejsze formatery (lata sześćdziesiąte) zorientowane były na jedno konkretne urządzenie wyjściowe i realizowały instrukcje prostych języków programowania automatycznego składu. Były to języki o bardzo skromnych możliwościach, a ich instrukcje opisywały elementarne czynności wykonywane w procesie formatowania bądź specyficzne parametry składu, jak „parametry pisma", „interlinia", „wcięcie akapitowe", „format wiersza (strony)" itp.

Współczesne języki formaterów, takich jak TEX (American Mathematical Society) czy NROFF/TROFF (Bell Laboratories), mają już wiele cech języków zaawansowanych, podobnych do występujących w tradycyjnych językach programowania, jak np. rozbudowane mechanizmy tworzenia i obsługi makroinstrukcji czy konstrukcje typu WHILE-DO, IF-THEN-ELSE. Wszystkie te systemy mają jedną wspólną cechę: instrukcje złożone konstruowane są za pomocą rozkazów elementarnych. W związku z tym ich użytkowników można podzielić na dwie grupy:
- użytkowników, którzy przy programowaniu składu korzystają z gotowych makroinstrukcji (ang. end users)
- użytkowników, którzy projektują i implementują makroinstrukcje (ang. layout designers).

Jedni i drudzy napotykają jednak w swojej pracy na pewne trudności. Pierwsi borykają się z brakiem logicznej spójności „wysokopoziomowych" instrukcji (metody ich tworzenia i reguły stosowania są dość złożone, a wynika to ze specyfiki automatycznego składu) oraz brakiem dobrej diagnostyki błędów (to również wynika ze specyfiki składu). Kłopoty drugich — to niski poziom języków używanych do tworzenia bardziej zaawansowanych konstrukcji oraz trudności w zwięzłym opisywaniu tak skomplikowanych form składu, jak: przypisy, tabele czy skład wielołamowy.

Ostatnio (lata osiemdziesiąte) niektórzy autorzy prezentują pomysły i projekty systemów w większym lub mniejszym stopniu wolnych od wspomnianych wyżej usterek. Jednym z takich systemów jest SCRIBE. Jego charakterystyczną cechą jest podejście parametryczne. Polega ono na opisywaniu klas dokumentów za pomocą ogromnej liczby (około stu) parametrów, tworząc w ten sposób dla danej klasy rodzaj szablonu pamiętanego w bazie danych. Definiowanie nowego rodzaju dokumentu ogranicza się do uaktualnienia bazy danych. System SCRIBE został zaprojektowany głównie z myślą o opracowaniach naukowych i technicznych.

Ciekawszym, a z pewnością też bardziej uniwersalnym systemem jest Strukturalny Formater Tekstu zaprojektowany w Centrum Matematycznym w Amsterdamie. Genezą jego powstania była również chęć uniknięcia kłopotów związanych ze stosowaniem dotychczas istniejących systemów. Twórcy systemu wytyczyli sobie dwa główne cele:

- zastąpienie znanych języków programowania automatycznego składu tekstu językiem wysokiego poziomu, umożliwiającym łatwe manipulowanie blokami tekstu
- realizację programową tzw. Abstrakcyjnej Maszyny Formatującej, która mogłaby być „mikroprogramowana" za pomocą tego języka; maszyny tej nie można utożsamiać z formaterem — konkretne formatery powstają właśnie przez mikroprogramowanie.

Dla ułatwienia, maszyna formatująca została zaimplementowana w tym samym języku programowania, który jest używany do jej mikroprogramowania. Reasumując: formater powstaje wskutek kompilacji maszyny formatującej razem ze źródłowym kodem jej mikroinstrukcji. Nie wdając się w szczegóły można powiedzieć, że Abstrakcyjna Maszyna Formatująca jest uogólnionym formaterem, wspólnym dla wszystkich możliwych klas dokumentów.

Realizuje ona następujące czynności:

- pobiera z wejścia dane źródłowe, tzn. tekst merytoryczny i proste rozkazy typu adiustacyjnego, definiujące wyróżnione bloki tekstu (tekst podstawowy, przypisy, tytuły itp.), rodzaj pisma oraz określające pewne niezbędne parametry numeryczne (np. formaty) i booleowskie (np. „z podziałem" lub „bez podziału na sylaby")
- dokonuje klasyfikacji jednostek tekstu dokumentu źródłowego
- wywołuje odpowiednie mikroprocedury, napisane w języku wysokiego poziomu, charakterystyczne dla danej klasy dokumentu.

Wykonanie tych procedur daje w efekcie określoną postać typograficzną. Warto wspomnieć, że istnieje możliwość definiowania przez użytkownika składni rozkazów adiustacyjnych. Opisany system został zaimplementowany na PDP11/45 pod nadzorem systemu operacyjnego UNIX, choć w uproszczonej, w stosunku do projektu, wersji. Językiem źródłowym — zarówno dla Abstrakcyjnej Maszyny Formatującej, jak i dla mikroprocedur — był SPRING, język przeznaczony do przetwarzania napisów. Problemem napotkanym podczas implementacji była zbyt mała, dostępna dla danych, pamięć operacyjna. Dlatego też wraz ze wzrostem rozmiaru stron rósł gwałtownie czas formatowania.

* * *

Należy sądzić, że rozwój systemów automatycznego przetwarzania danych tekstowych będzie następował wielokierunkowo. Na pewno dużą przyszłość mają przed sobą uniwersalne systemy, podobne do opisanego powyżej. Z drugiej strony trudno przypuszczać, aby popularność straciły proste, stosunkowo tanie automaty stosowane do bardzo specyficznych celów i używane coraz powszechniej w tzw. małej poligrafii. Wydaje się być również oczywiste, że z czasem będzie powiększał się repertuar używanych urządzeń wyjściowych.

LITERATURA

[1] Furuta R., Scofield J., Shaw A.: Document Formatting Systems: Survey, Concepts and Issues. Comp. Surveys, Vol. 14, No. 3, September 1982

[2] Knuth D. E.: TEX and METAFONT. Digital Press, 1979

[3] Noot H.: Structured Text Formatting. Software: Practice and Experience. Vol. 13, No. 1, Januar 1983

[4] Ossanna J. F.: NROFF/TROFF User's Manual. Bell Laboratories, Com. Science Technical Report 54, 1976

[5] Reid B. K.: SCRIBE, A Document Specification Language and its Compiler. Dep. of Comp. Science. Carnegie — Mellon University, October 1980.

KONKURS

Algorytm i program komputerowy rekonstrukcji obrazów tomograficznych metod CT i NMR

To temat konkursu ogłoszonego przez Zarząd Główny i Zarząd Sekcji Cybernetyki Medycznej Internistów Polskich.

Program może być uruchomiony na jednym z komputerów SM-4, RIAD-32, ODRA 1300, ICL 1900, MERA-400, MERA-50. Punktem startu programu może być macierz 262 × 262 danych początkowych, po translacjach, zapełniana sztucznie. Rozwiązaniem najlepszym jest uruchomienie programu na dowolnej, działającej w kraju instalacji tomografu komputerowego. Problem definiują wystarczająco dokładnie artykuły:

- Theory of Image Reconstruction in Computed Tomography, Radiology, 1975, 117, 561.
- Principles of Nuclear Magnetic Resonance Imaging, Radiology 1982, 143, 157.
- NMR Fourier zeugmatography. J. Magn. Reson., 1976, 18, 69.

Prace mogą być prowadzone w ramach tematów podejmowanych przez instytucje macierzyste autorów. Nagroda konkursowa uhonoruje jedynie wkład myśli osobistej autora algorytmu.

Wysokość nagród:

I nagroda — 50 000 zł
II nagroda — 20 000 zł
III nagroda — 10 000 zł

Zgłoszenia gotowości do demonstracji działania programu należy zgłaszać do dnia 1.11.1984 r. pod adresem Zarządu Sekcji Cybernetyki Medycznej Towarzystwa Internistów Polskich — V Katedra i Klinika Chorób Wewnętrznych Śląskiej Akademii Medycznej, ul. Żeromskiego 7, 41-902 Bytom, tel. 810-231.

Nagrody przyzna Sąd Konkursowy, a ogłoszenie wyników nastąpi w czasopismach lekarskich i ogólnych w lutym 1985 roku.

Raport dla Klubu Rzymskiego: Mikroelektronika i społeczeństwo
Na dobre i złe (3)

Po wydrukowaniu pełnego rozdziału Raportu poświęconego wpływowi komputeryzacji na zatrudnienie (nr 5, 6/83), przedstawiamy obecnie fragmenty innego rozdziału, dotyczącego powiązań mikroelektroniki z rozwojem techniki wojskowej, a więc — wyścigiem zbrojeń. Wybór tej części jest podyktowany znaczeniem, jakie mikroelektronika, a zatem i informatyka, ma dla rozwoju technologii.

Zastosowania mikroelektroniki w urządzeniach wojskowych określają perspektywę rozwoju całej informatyki. Niezależnie od oceny owej „militaryzacji" informatyki, fakty zdają się być oczywiste: komputery służą zbrojeniom — i odwrotnie. Państwa przodujące w rozwoju militariów skrzętnie ukrywają własne zdobycze technologii. Niemniej o skali przemian świadczą rodzaje zastosowań, o których dzisiaj — w wyniku głośnej kampanii rozbrojeniowej — wiemy coraz więcej. Raport ogranicza się w zasadzie do omówienia elementów uzbrojenia USA, nie siląc się zatem na przedstawienie pełnego obrazu światowych zbrojeń. Ale już poniższe fakty będą dla informatyka dostatecznie wymowne. (Red.)

FRANK BARNABY
Stockholm International Peace Research Institute
Hampshire, Wielka Brytania

Zastosowanie mikroelektroniki w urządzeniach wojskowych

Opis wszystkich zastosowań mikroelektroniki w wojsku zająłby wiele tomów. Poniżej omówimy więc jedynie cztery rodzaje zastosowań: w pociskach strategicznych i sterowanych, samolotach wczesnego ostrzegania, automatyzacji działań bojowych i nowoczesnych samolotach bojowych. Wybrano je tak, by przedstawić zakres wykorzystania mikroelektroniki w wojsku oraz dalsze konsekwencje w taktyce i strategii wojskowej, w przyszłych działaniach bojowych, a wreszcie w rozwoju zbrojeń jądrowych.

WOJSKOWE PROGRAMY BADAWCZE

Na badania wojskowe wydaje się ogromne ilości pieniędzy, których duża część przeznaczona jest na rozwój wyposażenia elektronicznego systemów broni. W latach sześćdziesiątych na badania wojskowe wydatkowano średnio ok. 16 mld dol. rocznie, tj. około 10% światowych wydatków zbrojeniowych. Obecnie na badania wojskowe wydaje się z grubsza 50 mld dol. rocznie, co stanowi również około 10% wszystkich wydatków na zbrojenia. Nawet te olbrzymie sumy wynikają z zaniżonych ocen.

W cenach stałych, uwzględniając inflancję, światowe wydatki na badania wojskowe zwiększyły się o ok. 60% w latach sześćdziesiątych i ok. 20% w latach siedemdziesiątych. Oceniając planowane przez niektóre kraje wydatki na badania wojskowe, można przypuszczać, że w latach osiemdziesiątych będą one wzrastać jeszcze szybciej niż w ubiegłym dziesięcioleciu.

W ubiegłym dwudziestoleciu zdecydowaną większość badań wojskowych prowadzono w Stanach Zjednoczonych i Związku Radzieckim. Te dwa kraje wydatkują ok. 80% wszystkich pieniędzy przeznaczonych na ten cel. Francja, Wielka Brytania, RFN i Chiny wydają łącznie ok. 20% tego, co Stany Zjednoczone. Pozostałe kraje wydają nie więcej niż 5% całej sumy przeznaczonej na te badania. Do najbardziej znaczących krajów tej grupy należą: Australia, Holandia, Indie, Japonia, Kanada, Szwecja i Włochy.

POCISKI STRATEGICZNE

Najbardziej dramatycznymi skutkami postępu elektroniki są te, które czynią bardziej prawdopodobną powszechną wojnę jądrową. Zaliczamy do nich zwłaszcza poprawę celności i niezawodności pocisków strategicznych, a także rozwój powietrznych środków nawigacji, ulepszanie technik zwalczania okrętów podwodnych, rozwój metod niszczenia głowic nieprzyjacielskich pocisków balistycznych w przestrzeni i rozwój metod niszczenia nieprzyjacielskich satelitów.

Choć strategiczna broń jądrowa jest wycelowana w obiekty wojskowe, oficjalna polityka w dziedzinie zbrojeń jądrowych opiera się na wzajemnym zagrożeniu zniszczeniem przeciwnika. Teoria ta polega na założeniu, że nieprzyjaciel nie zaatakuje jako pierwszy, jeżeli w odwet zostaną zniszczone jego miasta i przemysł.

Im bardziej dwa największe mocarstwa dostosowują swoje doktryny do prowadzenia wojny jądrowej, tym większe jest prawdopodobieństwo jej rozpoczęcia, ponieważ będzie więcej podstaw do stwierdzenia, że prowadzenie i wygranie takiej wojny jest możliwe.

W odpowiednim czasie mogą powstać ofensywne i defensywne systemy strategicznej broni jądrowej, które

umożliwią niespodziewany, obezwładniający atak lub — zdaniem niektórych naukowców — uczynią go prawdopodobnym, a nawet — nieuniknionym.

Znacznie zwiększy się także możliwość wybuchu powszechnej wojny jądrowej przez przypadek lub omyłkę w obliczeniach, np. wskutek błędu w technologii lub mylnej interpretacji danych.

Dotychczas zdarzyło się wiele fałszywych alarmów spowodowanych przez amerykańskie komputery wojskowe. Choć alarmy te nie były skutkiem rzeczywistego ataku jądrowego, stwierdzono to dopiero po 15—20 minutach. Właśnie z tego powodu planowane rozmieszczenie w Republice Federalnej Niemiec pocisków Pershing II, wycelowanych w europejską część Związku Radzieckiego, znacznie zwiększy niebezpieczeństwo przypadkowego wybuchu wojny jądrowej. Spowodowanie wojny jądrowej stanie się możliwe przez pociągnięcie spustu.

POCISKI MANEWRUJĄCE TYPU CRUISE

Nowoczesne pociski typu CRUISE są systemem strategicznej broni jądrowej, opartym głównie na mikroelektronice. Najważniejsza, jak dotychczas, okazała się **miniaturyzacja komputerów**, tzn. zmniejszenie ich objętości i ciężaru, przy określonej mocy obliczeniowej. Równie ważny stał się dostęp do bardzo dokładnych danych dotyczących współrzędnych potencjalnych celów. Dzięki temu mogły powstać bardzo małe, lecz wystarczająco dokładne, systemy kierowania pociskami. Przykładowo — system dopasowania do kształtu terenu (TERCOM) firmy McDONNEL DOUGLAS waży zaledwie 37 kg i może naprowadzić pocisk typu CRUISE na cel z promieniem błędu trafienia równym kilkadziesiąt metrów.

W systemie TERCOM użyto pokładowego komputera do porównywania kształtu terenu pod pociskiem (na podstawie pomiaru dokonanego wysokościomierzem radarowym) z zaprogramowaną trasą przelotu. Odchylenia od planowanej trasy przelotu są automatycznie korygowane. Położenia celów i kształt terenu, na trasie przelotu, można określić z niespotykaną dokładnością na podstawie bardzo dokładnych map uzyskanych techniką zdjęć satelitarnych. Na podstawie dotychczasowych map nie można było określić położenia celów z dokładnością dostateczną do skutecznego użycia systemów kierowania pociskami typu CRUISE. Te systemy kierowania są oparte na rozpoznawaniu obrazów, jednej z głównych — wykorzystywanych w wojsku — **metod sztucznej inteligencji.**

Przy użyciu nowych technologii, w Stanach Zjednoczonych konstruowane są pociski typu CRUISE wystrzeliwane z lądu, morza i powietrza. Chyba najważniejszą ich cechę określa fakt, że stosunek przenoszonego ładunku do fizycznego ciężaru pocisku jest względnie duży, równy ok. 15% w porównaniu z 1% dla typowych pocisków balistycznych.

Przenoszone na bombowcach strategicznych i wystrzeliwane z powietrza pociski typu CRUISE mają bardzo mały obraz radarowy i latają z szybkością poddźwiękową na bardzo małych wysokościach, rzędu kilkuset metrów nad terenem nierównym i kilkudziesięciu metrów nad terenem płaskim. Pocisk w locie jest bardzo trudny do wykrycia i zniszczenia, dlatego obrona przeciwrakietowa jest trudna i kosztowna; szczególnie, jeżeli wystrzelono dużą liczbę pocisków nieprzyjacielskich. Każdy amerykański bombowiec strategiczny będzie przenosić do 24 pocisków typu CRUISE wyposażonych w głowice jądrowe.

Skuteczne wykrywanie wystrzeliwanych z powietrza pocisków typu CRUISE będzie prawdopodobnie wymagało obserwacji radarem zainstalowanym na samolocie systemu wczesnego ostrzegania (AWACS). Do patrolowania długiej granicy konieczne byłoby użycie całej floty powietrznej, co jest przedsięwzięciem bardzo kosztownym. Niezbędna byłaby ponadto duża liczba samolotów pościgowych dalekiego zasięgu, do współpracy z systemem AWACS w przechwytywaniu i niszczeniu nadlatujących pocisków, co również wymaga ogromnych nakładów. Dlatego też rozprzestrzenienie pocisków typu CRUISE może spowodować dalszą eskalację wyścigu zbrojeń.

System obrony przeciwko pociskom typu CRUISE jest na ogół bardziej kosztowny od samych pocisków. Pomimo dużej celności i względnej doskonałości, pociski typu CRUISE są dosyć tanie. W serii produkcyjnej, powiedzmy — 2000 pocisków, koszt jednostkowy z uwzględnieniem kosztu opracowania może wynosić ok. 750 tys. dol., tj. znacznie mniej niż koszt nowoczesnego czołgu bojowego.

Jeżeli pociski typu CRUISE rozpowszechnią się, to mogą okazać się najbardziej brzemiennym w następstwa osiągnięciem technologii wojskowej. Nie ma wątpliwości, że ulepszone odmiany pocisków typu CRUISE będą w przyszłości powstawać szybko po sobie.

SYSTEM WCZESNEGO OSTRZEGANIA — AWACS

System AWACS jest nadzwyczaj kosztownym składnikiem uzbrojenia opartego na nowoczesnej technologii, złożonym z dużej liczby **urządzeń mikroelektronicznych.** Projekt najnowszego amerykańskiego samolotu systemu AWACS, typ E-3A, kosztuje 1,5 mld dol., a koszt wyprodukowania każdego egzemplarza wynosi 100 mln dol., tj. 6 razy tyle, co najbardziej skomplikowanego samolotu bojowego.

Wielu specjalistów uważa samoloty typu AWACS za mikroelektroniczne słonie, odkryty i łatwy cel — możliwy do strącenia zaraz na początku wojny. Ponadto samolot systemu AWACS będzie musiał, co oczywiste, operować w środowisku znacznego przeciwdziałania radioelektronicznego. Oponenci zaś twierdzą, że radar jest niedostatecznie skutecznym środkiem kontrprzeciwdziałania.

Zwolennicy systemu AWACS argumentują, że jest on bardziej odporny na przeciwdziałanie radioelektroniczne, niż obecne, naziemne, przeszukiwawcze stacje radiolokacyjne i że przetrwa w nie sprzyjającym środowisku wystarczająco długo, aby odegrać znaczną rolę w działaniach bojowych. Podkreśla się także znaczenie ostrzegawcze wczesnego, choć krótkiego okresu na początku konfliktu.

Nowoczesny samolot systemu AWACS, jak np. wspomniany już E-3A, wykonuje wszystkie funkcje samolotu wczesnego ostrzegania i dodatkowo spełnia wiele funkcji dowodzenia i sterowania wszystkimi własnymi samolotami pozostającymi w jego zasięgu, a w tym — samolotami pościgowymi, transportowymi, zwiadowczymi itp. Skuteczność identyfikacji i śledzenia samolotów nieprzyjacielskich i sterowania własnymi — na dużych odległościach, małych i dużych wysokościach, w każdych warunkach pogodowych, nad lądem i morzem — osiągnięto dzięki zastosowaniu bardzo skomplikowanych, sięgających poza horyzont, obserwacyjno-przeszukiwawczych impulsowych urządzeń radarowych wielkiej częstotliwości — opartych na zjawisku Dopplera, bardzo szybkich komputerów i wielofunkcyjnych monitorów ekranowych.

Samolot typu E-3A jest zmodyfikowaną maszyną BOENING 707. Chłodzone płynem anteny radarowe są umieszczone na obrotowej platformie, zamocowanej na kadłubie, poruszanej hydraulicznie z prędkością 6 obrotów na minutę. Zakres przeszukiwania obejmuje obszar od poziomu ziemi do wysokości stratosferycznych.

Ze względu na długi czas patrolowania, które może trwać do 11 godzin, a po uzupełnieniu paliwa w powietrzu — nawet ponad 20, samoloty typu E-3A mają bardzo dokładny system nawigacji. W każdym samolocie, oprócz czterech osób załogi kierującej lotem, może znajdować się około dwunastu innych specjalistów.

AUTOMATYZACJA DZIAŁAŃ BOJOWYCH

Na szczeblu taktycznym, mikroelektronika ma największy udział w automatyzacji działań bojowych. W ubiegłych piętnastu latach poczyniono znaczne postępy w dziedzinie automatyzacji walki na ziemi, morzu i w powietrzu. Jest to chyba najdynamiczniej rozwijająca się dziedzina zastosowań technologii wojskowych, a jednocześnie — otoczona największą tajemnicą.

W większości automatycznych systemów przeznaczonych do działań bojowych stosuje się mikroelektronikę. Rozwój nowych, elektronicznych broni ofensywnych powoduje nieuchronny rozwój elektronicznych środków przeciwdziałania. To, z kolei, wywołuje rozwój elektronicznych środków kontrprzeciwdziałania itd. Nie powinno nikogo dziwić, że wojskowe rewolucje mikroelektroniczne następują po sobie w oszałamiającym tempie. W tej dziedzinie nikt nie może oczekiwać, że dorówna wszystkim najlepszym rozwiązaniom, nawet gdyby otrzymał odpowiednie informacje.

W większości działań bojowych wyróżnia się cztery fazy. To pierwsze — lokalizuje się i identyfikuje siły nieprzy-

jaciela; po drugie — podejmuje się decyzje co do sposobu prowadzenia walki; po trzecie — kieruje się na przeciwnika odpowiednie rodzaje ognia; wreszcie — po czwarte — ocenia się straty nieprzyjaciela, w celu ewentualnego powtórzenia całej sekwencji.

Przy automatyzacji działań bojowych, siły i środki nieprzyjaciela byłyby wykrywane przez zdalnie sterowane pojazdy lub czujniki zainstalowane na ziemi. Zebrane w ten sposób dane byłyby przesyłane do komputerów, a te podejmowałyby decyzje co do dalszych działań i skierowania ognia na dany cel. Po przerwaniu ognia czujniki rozmieszczone na obszarze działań i zdalnie sterowane pojazdy dokonałyby oceny zniszczeń i skierowałyby te informacje do komputerów, które podjęłyby decyzje co do dalszego prowadzenia ognia itd.

Generał William Westmoreland, były Szef Sztabu Sił Lądowych USA, przewidywał, że „w przyszłych działaniach bojowych, siły nieprzyjaciela będą lokalizowane, śledzone i brane na cel prawie natychmiast, dzięki użyciu łączy cyfrowych, komputerowo wspomaganemu rozpoznawaniu i automatycznemu sterowaniu ogniem. Przy prawdopodobieństwie zniszczenia w pierwszym uderzeniu zbliżonym do pewności i istnieniu urządzeń przeszukiwawczych, śledzących nieprzyjaciela w sposób ciągły, potrzeba posiadania znacznych sił do fizycznego zdławienia oporu jest mniej istotna". Na podstawie dostępnych danych wydaje się prawdopodobne, że automatyzacja działań bojowych przewidywana przez generała stanie się realna w latach osiemdziesiątych.

WYPOSAŻENIE NOWOCZESNEGO LOTNICTWA

Nowoczesny samolot bojowy jest doskonałym przykładem systemu broni, której działanie opiera się na najnowszych osiągnięciach elektroniki. Najnowocześniejszym zastosowaniem elektroniki w lotnictwie są samoloty myśliwskie przeznaczone do poszukiwania, wykrywania i niszczenia wszelkich typów samolotów nieprzyjacielskich, w dowolnych warunkach pogodowych. Samoloty tego typu mogą wykonywać również inne zadania, np. atak z powietrza.

Typowy nowoczesny samolot bojowy zawiera lekki system radarowy, służący do wykrywania i śledzenia szybko poruszających się obiektów na dużych odległościach i na wszystkich wysokościach, aż do poziomu wierzchołków drzew. Informacja o śledzonym obiekcie jest zwykle wprowadzana do komputera pokładowego, który na tej podstawie określa kierunek wystrzeliwanego pocisku. W przypadku walki bezpośredniej, radar zapewnia automatyczne zobrazowanie obiektu, z wyświetlaniem w postaci symboli całej niezbędnej informacji — na ekranie monitora, umieszczonego na wysokości oczu pilota. Pilot otrzymuje więc automatycznie dane niezbędne do przechwycenia i zniszczenia nieprzyjacielskiego samolotu, sam nie spuszczając go z oka. Monitor ekranowy dostarcza pilotowi również wszelkich informacji niezbędnych do nawigacji i sterowania w każdych warunkach, ze wszystkimi szczegółami dotyczącymi działania samolotu — tak, aby można wykryć dowolne uszkodzenie w dowolnym podsystemie samolotu, gdy tylko ono nastąpi.

System identyfikacji obiektów informuje pilota, czy samolot wykryty wzrokiem lub radarem jest własny czy nieprzyjacielski. Natomiast komputer analizujący warunki lotu oraz zestaw urządzeń do kontroli orientacji i położenia — wyświetlają informacje dotyczące pochylenia oraz kierunku linii pola magnetycznego. Wszystko to, w połączeniu z inercyjnym przyrządem do nawigacji, umożliwia pilotowi kierowanie samolotem w każdych warunkach.

Stany Zjednoczone rozważają wprowadzenie do uzbrojenia tzw. samolotu pościgowego szybkiego reagowania, rozwijającego prędkość trzykrotnie większą od prędkości rozchodzenia się dźwięku (3 Ma) i niezwykle duże przyspieszenia, przeznaczonego do działania w przyszłym systemie powietrznej broni defensywnej. Oprócz funkcji samolotu pościgowego, taki samolot może pełnić funkcje samolotu myśliwskiego i bojowego. Poważnie rozważa się także możliwość wprowadzenia jeszcze szybszych samolotów ponaddźwiękowych. Budowa tak doskonałych samolotów bojowych jest możliwa głównie dzięki doskonałemu wyposażeniu elektronicznemu.

Typowy samolot myśliwski zostanie uzbrojony w pociski typu powietrze-powietrze, bliskiego i średniego zasięgu. Przykładowymi pociskami wielozadaniowymi, typu powietrze-powietrze, dostosowanymi do różnych warunków pogodowych są amerykańskie pociski typu SPARROW. Są one wyposażone w półaktywny, radarowy system naprowadzania, oparty na fali ciągłej i przenoszą pięciokilogramowe głowice o dużej sile rażenia, wybuchające w pobliżu celu. Rakieta na paliwo stałe wyrzuca pocisk z prędkością 3 Ma, na odległość ok. 50 km.

Wyposażenie pocisku jest zminiaturyzowane, a konsekwentna redukcja jego rozmiarów umożliwia zainstalowanie większego silnika na paliwo stałe. Pocisk ma także duże możliwości przeciwdziałania elektronicznego, jest niezawodny i manewrowalny w bezpośredniej walce.

W systemie namierzania dla najnowszych pocisków powietrze-powietrze, typu SPARROW, połączono urządzenia radarowe i oparte na wykorzystaniu promieniowania podczerwonego. Pociski z półaktywnym, radarowym systemem naprowadzania wymagają ciągłego opromieniowywania celu. Radar samolotu emituje ciągły sygnał opromieniujący — tak, że po wystrzeleniu pocisk może kierować się odbiciami od celu. Ponieważ antena radaru samolotowego musi być skierowana na cel aż do trafienia go przez pocisk, w tym czasie nie można obserwować innych obiektów. Natomiast pociski wykorzystujące promieniowanie cieplne do samonaprowadzania na cel wystarczy jedynie wystrzelić w jego kierunku.

Tłumaczył JANUSZ ZALEWSKI

KONFERENCJE

Mikroprocesory w automatyce i pomiarach

Sekcja Automatyki i Pomiarów Oddziału Warszawskiego Elektroniki i Telekomunikacji SEP organizuje **IV Krajową Konferencję Naukowo-Techniczną pn. „Zastosowanie mikroprocesorów w automatyce i pomiarach"**. Konferencja poświęcona będzie:

— zastosowaniom mikroprocesorów w układach sterowania i aparaturze pomiarowo-kontrolnej
— zastosowaniom układów kalkulatorowych w automatyce i pomiarach
— podzespołom, blokom funkcjonalnym, urządzeniom pomocniczym i oprogramowaniu sterowników mikroprocesorowych
— problemom projektowania, uruchamiania i diagnostyki mikroprocesorowych urządzeń automatyki i pomiarów.

Konferencja odbędzie się w Warszawie w dniu 20 września 1984 r. Zgłoszenia referatów i komunikatów będą przyjmowane do 30.12.1983 i zostaną rozpatrzone w styczniu 1984 r. Zakwalifikowane wystąpienia, których teksty zostaną nadesłane do organizatorów do dnia 30.03.1984 będą opublikowane w materiałach konferencyjnych, doręczonych wszystkim uczestnikom. Program konferencji zostanie ustalony i rozesłany uczestnikom w czerwcu 1984 r.

Wstępne zgłoszenia uczestnictwa (z tytułami i kilkuwierszowym streszczeniem oraz określeniem objętości proponowanych referatów i komunikatów) prosimy nadsyłać pod adresem sekretarza organizacyjnego:

mgr Lech Szyngwelski
Zakłady Urządzeń Elektronicznych „Unitra-Unima"
ul. Konstruktorska 9,
02-673 Warszawa; (tel. 43-62-21 w. 232)

ZYGMUNT LISZYŃSKI
Środowiskowy Ośrodek Informatyki
Politechnika Poznańska

Programowanie procesów współbieżnych (1)

Środowisko sprzętowe

Artykuł ten stanowi pierwszą część publikacji omawiających wyniki prac prowadzonych w Środowiskowym Ośrodku Informatyki PP nad utworzeniem środowiska sprzętowo-programowego umożliwiającego programowanie procesów współbieżnych. Druga część (nr 10/83) przedstawi środowisko programowe, a przede wszystkim — opis struktury hierarchii maszyn wirtualnych, tworzonych przez kolejne programy zarządzające.

Wspomniane środowisko sprzętowo-programowe utworzono dla maszyny cyfrowej ODRA 1305, wykorzystując jej program zarządzający Executive E6RM. Wiele z zaproponowanych rozwiązań układowych i programowych można wykorzystać przy konstruowaniu podobnego środowiska dla innych maszyn cyfrowych, zwłaszcza — dla maszyn, które wykorzystują technikę rejestrów granicznych dla podziału i ochrony pamięci operacyjnej.

Badanie metod programowania procesów współpieżnych podjęto w związku z prowadzonymi od połowy lat siedemdziesiątych pracami nad realizacją jednodziedzinowych, wielodostępnych systemów informatycznych. W systemach takich procesy obliczeniowe są klasami jednorodne oraz wzajemnie związane. Jednorodność procesów umożliwia reprezentowanie identycznego dla nich algorytmu w postaci współużytkowanego oprogramowania oraz wspólnych danych stałych. Wzajemne wiązanie procesów odbywa się poprzez udostępnienie współużytkowanych zmiennych (danych aktualizowanych) oraz odpowiednich środków synchronizacji. Specyficzne dla każdego procesu zmienne tworzą zbiór jego obiektów lokalnych.

Dla efektywnej realizacji jednodziedzinowych systemów informatycznych utworzono wyspecjalizowane środowisko sprzętowo-programowe. Właściwości tego środowiska są na tyle uniwersalne, że umożliwia ono realizację dowolnego systemu procesów współbieżnych. Środowisko to tworzą: maszyna cyfrowa ODRA 1305 o zmodyfikowanym procesorze centralnym (wyposażona w procesor komunikacyjny i pamięć dyskową), zmodyfikowany program zarządzający Executive E6RM, dwupoziomowy system operacyjny MJOS/INZA oraz kompilator języka PLAN 5, zaprojektowanego dla programowania procesów współbieżnych.

Dr inż. ZYGMUNT LISZYŃSKI ukończył w 1969 r. studia na Wydziale Elektrycznym Politechniki Poznańskiej. Od roku 1969 do chwili obecnej pracuje w Środowiskowym Ośrodku Informatyki Politechniki Poznańskiej. W roku 1978 obronił pracę doktorską z zakresu interakcyjnych systemów operacyjnych. Zajmuje się zagadnieniami organizacji maszyn i systemów cyfrowych, w szczególności problemami konstrukcji krotnych systemów mikroprocesorowych i oprogramowania zarządzającego.

Przystępując do projektowania omawianego środowiska sformułowano wymagania, jakie spełnić musi sprzęt komputerowy. Podstawowym wymogiem jest tu jego przystosowanie do użytkowania wielodostępnego. Dla realizacji systemów jednodziedzinowych istotne jest współużytkowanie oprogramowania i niektórych danych przez wiele procesów obliczeniowych. Ze względów użytkowych dąży się przy tym do jednoczesnego spełnienia dwóch postulatów:
— opisu algorytmu dla określonej klasy zadań jednorodnych w sposób niezależny od mocy tej klasy
— realizacji tej właściwości na niższym poziomie hierarchii systemu informatycznego niż poziom procesorów użytkowych. Drugi z tych postulatów oznacza wymaganie, aby konstruktor oprogramowania nie musiał respektować specjalnych ograniczeń w zasadach programowania, wyrażających się w fakcie istnienia zmiennych współużytkowanych z podsystemem zarządzania (w przeciwieństwie do zmiennych współużytkowanych z innymi związanymi procesami obliczeniowymi).

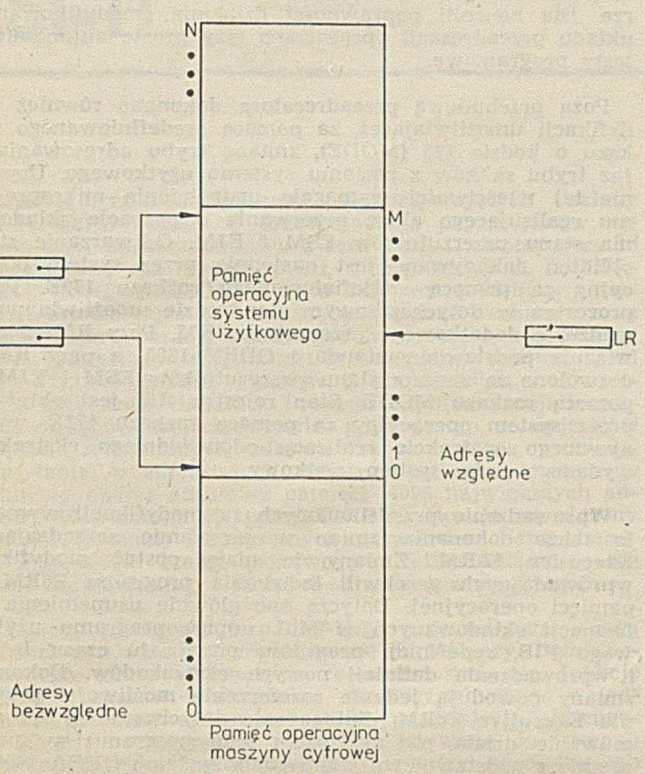
Wymienione wymagania mogą być łatwo spełnione w systemach komputerowych, w których wprowadzono technikę segmentacji. Inaczej jest, gdy — jak w maszynach ODRA 1305 — wykorzystuje się technikę rejestrów granicznych. Technika ta zasadniczo nie pozwala na bezpośrednie współużytkowanie danych i oprogramowanie przez różne procesy obliczeniowe. Dlatego rozwiązanie tego problemu warunkowało racjonalną implementację jednodziedzinowego systemu informatycznego dla ODRY 1305.

ROZWINIĘTE METODY ADRESOWANIA

Rozważając właściwości techniki rejestrów granicznych można zauważyć, że wymaga ona zespolenia w jednym, przydzielonym systemowi użytkowemu, segmencie pamięci operacyjnej różnych rodzajów obiektów, takich jak: dane lokalne, dane globalne, oprogramowanie, bufory transmisyjne itp. Klasyczny wariant tej techniki można więc nazwać przeadresacją zespoloną (przeadresacja — translacja adresów) (rys. 1). Obszar segmentu wyznaczony jest przez dwie liczby adresowe zawarte w rejestrach graficznych — dolnym (D) i górnym (G). Adresy względne danych wymagają przeliczenia na adresy bezwzględne pamięci operacyjnej. Licznik rozkazów (LR) zawierać może adres względny albo bezwzględny. W maszynie ODRA 1305 znajduje się w nim adres bezwzględny, dzięki czemu przeliczanie adresu rozkazu odbywa się tylko przy realizacji rozkazów skokowych i przy składowaniu i odtwarzaniu zawartości licznika rozkazów.

Rozwiązaniem opartym na odmiennych przesłankach w stosunku do przeadresacji zespolonej jest przeadresacja rozłączna. Dla każdego rodzaju wykorzystywanych przez system użytkowy obiektów wyznacza się odrębny segment pamięci operacyjnej. Jako odrębne rodzaje obiektów wyróżniono: zmienne semaforowe, dane współużytkowane

zwane globalnymi, współużytkowane, inwariantne oprogramowanie oraz dane lokalne procesu obliczeniowego. Dla wyróżnienia czterech segmentów pamięci operacyjnej, jakie w przypadku przeadresacji rozłącznej wykorzystuje system użytkowy, konieczne są cztery pary rejestrów granicznych. Należy zauważyć, że wprowadzenie przeadresacji rozłącznej nie jest powrotem do techniki segmentacji, bowiem cztery wyróżnione w tym rozwiązaniu segmenty pamięci operacyjnej są segmentami wyspecjalizowanymi, natomiast w przypadku klasycznej segmentacji tworzone są segmenty funkcjonalnie jednorodne (różniące się być może jedynie wielkością).

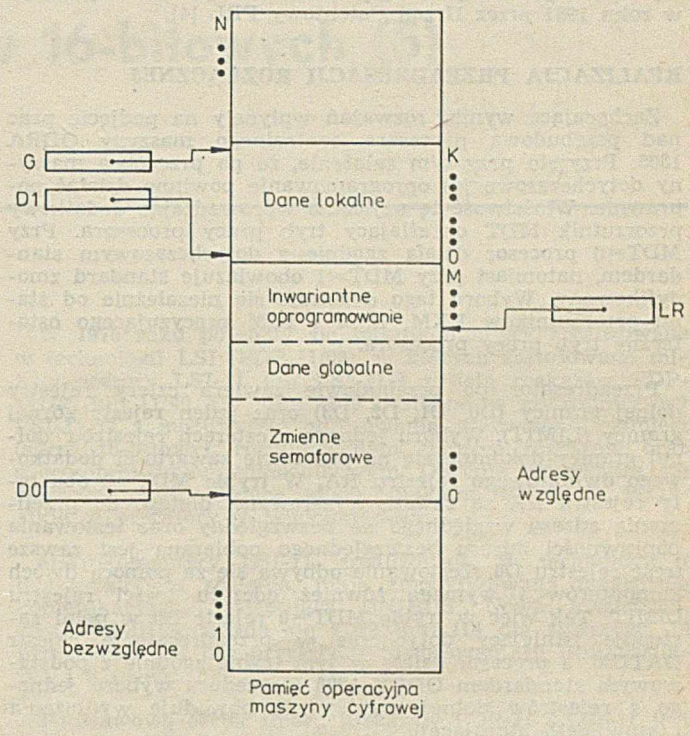


Rys. 1. Przeadresacja zespolona

Wprowadzenie czterech odrębnych segmentów wymaga pozyskiwania informacji dodatkowej, umożliwiającej przy każdym odwołaniu wybór jednej z czterech par rejestrów granicznych. Możliwe są tu dwa rozwiązania. Po pierwsze — do wskazywania jednego z czterech segmentów można wykorzystać dwa bity adresu względnego. Rozwiązanie to powoduje czterokrotne skrócenie obszaru adresowanego adresem pierwotnym (niemodyfikowanym). Druga możliwość — to wprowadzenie dla każdego segmentu oddzielnego zbioru rozkazów. Zestawy rozkazów służących do współpracy z segmentem zmiennych semaforowych i segmentem danych globalnych mogą być stosunkowo nieliczne. W skrajnym przypadku wystarczają dwa rozkazy realizujące operacje P i V na zmiennych semaforowych oraz rozkazy odczytu i zapisu z segmentów danych globalnych. Rozkazy przetwarzające zmienne semaforowe powinny — gdy jest to konieczne — powodować przerwanie, umożliwiające systemowi operacyjnemu organizację biernego oczekiwania. Kontakt z segmentem oprogramowania zachodzi natomiast jedynie w podcyklu pobrania rozkazu. Podstawowa część listy rozkazów dotyczy w tym rozwiązaniu segmentu danych lokalnych.

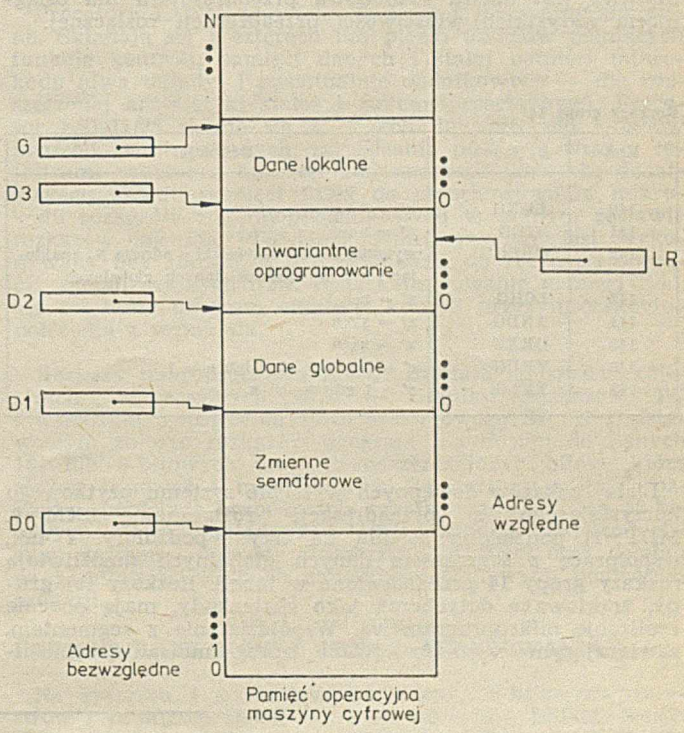
W przypadku przeadresacji rozłącznej przeadresator zawiera, poza sumatorem i dwoma komparatorami, osiem rejestrów granicznych oraz odpowiednie układy wybierające jedną z czterech par tych rejestrów. Uproszczenie budowy przeadresatora można uzyskać uwzględniając pośrednie warianty techniki rejestrów granicznych. Godne odnotowania są tu dwa warianty rozwiązań, w których przyjmuje się założenie, że system użytkowy dysponuje spójnym, chociaż wielosegmentowym obszarem pamięci. Obszar ten z punktu widzenia układu ochrony informacji stanowi jedną całość.

Pierwszy z wariantów, przedstawiony na rysunku 2, uzyskano dokonując częściowego zespolenia współużytkowanych obiektów różnych typów w jednym segmencie pamięci. Dane lokalne umieszczone są w odrębnym segmencie.



Rys. 2. Przeadresacja częściowo zespolona

W drugim wariancie (rys. 3) początki różnych segmentów wyznaczane są przez własne adresy odniesienia. Każdy z czterech typów obiektów rozmieszczony jest w odrębnym segmencie.



Rys. 3. Uproszczone rozwiązanie przeadresacji rozłącznej

W obu przedstawionych wariantach następuje pogorszenie zdolności wykrywania niepoprawnych odwołań do pamięci. Fakt ten nie ma jednak istotnego znaczenia podczas eksploatacji dobrze zweryfikowanych systemów informatycznych.

Przedstawione metody adresacji pamięci w maszynach z rejestrami granicznymi są przedmiotem patentu wydanego w roku 1981 przez Urząd Patentowy PRL [4].

REALIZACJA PRZEADRESACJI ROZŁĄCZNEJ

Zachęcające wyniki rozważań wpłynęły na podjęcie prac nad przebudową procesora centralnego maszyny ODRA 1305. Przyjęto przy tym założenie, że po przeróbce maszyny dotychczasowe jej oprogramowanie powinno działać poprawnie. Właściwość tę uzyskano wprowadzając dodatkowy przerzutnik MDT określający tryb pracy procesora. Przy MDT=0 procesor działa zgodnie z dotychczasowym standardem, natomiast przy MDT=1 obowiązuje standard zmodyfikowany. Wyboru tego dokonuje się niezależnie od stanu przerzutników EXM, ESM i EJM precyzującego ostateczny tryb pracy procesora.

Przeadresator po przebudowie zawiera cztery rejestry dolnej granicy (D0, D1, D2, D3) oraz jeden rejestr górnej granicy (LIMIT). Wyboru jednego z czterech rejestrów dolnej granicy dokonuje się na podstawie zawartości dodatkowego dwubitowego rejestru RA. W trybie MDT=0 oba bity rejestru RA są zawsze wyzerowane, dlatego do przeliczenia adresu względnego na bezwzględny oraz testowania poprawności adresu bezwzględnego pobierana jest zawsze treść rejestru D0. Testowanie odbywa się za pomocą dwóch komputerów i wymaga również odczytu treści rejestru LIMIT. Tak więc w trybie MDT=0 rejestr D0 w pełni zastępuje istniejący dotychczas w przeadresatorze rejestr DATUM, a procesor działa w tym trybie zgodnie z podstawowym standardem ODRY 1305 (procedura wyboru jednego z rejestrów dolnej granicy nie powoduje wydłużenia czasów pętli sterującej).

W trybie MDT=1 obowiązuje uproszczona wersja przeadresacji rozłącznej (rys. 3). Rejestry dolnych granic określają początki obszarów zmiennych semaforowych, danych globalnych, współużytkowanego oprogramowania oraz danych lokalnych. Współpracę z każdym z tych obszarów zapewnia odrębny zestaw rozkazów, dlatego z definicją poszczególnych grup rozkazów związana jest wartość, jaka wpisywana jest do rejestru RA. W zrealizowanej wersji przeadresacji rozłącznej jako granicę górną przyjmuje się dla wszystkich odwołań treść rejestru LIMIT. Niemniej możliwa jest dalsza rozbudowa przeadresatora dla osiągnięcia wszystkich właściwości przeadresacji rozłącznej.

Rozkazy grupy 14

Kod	Nazwa	Definicja
140	LDXG	$x' = n+c$
141	STOG	$n' = x+c$
142	OBEG	wykonaj rozkaz z komórki o adresie N_e znajdującej się w obszarze danych globalnych
143	LCHG	$x' = n_j$
144	ANDG	$x' = x \wedge n$
145	ORXG	$x' = x \vee n$
146	TXUG	$c' = 1$ gdy $n \neq x$ lub $c = 1$
147	TXLG	$c' = 1$ gdy $n+c>x$

Lista rozkazów dostępnych w trybie systemu użytkowego korzystającego z przeadresacji rozłącznej (EXM=0, MDT=1) podzielona została na trzy podzbiory. I tak, współpracę z segmentem danych globalnych umożliwiają rozkazy grupy 14 przedstawione w tabeli. Rozkazy tej grupy, traktowane dotychczas jako ekstrakody, mają obecnie realizację mikroprogramową. Współdziałanie z segmentem, zawierającym współużytkowane oprogramowanie, umożliwiają rozkazy skokowe (grupy 05, 06 i 07). Wszystkie pozostałe rozkazy dotyczą segmentu danych lokalnych. Nie zrealizowano dotychczas mikroprogramowych rozkazów wspomagających realizację operacji P i V na zmiennych semaforowych, dlatego operacje te realizowane są ekstrakodowo. Przełączenie procesora z trybu MDT=0 na MDT=1 i odwrotnie oraz ustalanie treści rejestrów dolnych granic i rejestru górnej granicy odbywa się za pomocą zmodyfikowanych rozkazów 172E i 173E.

Modyfikacji procesora dokonano zastępując pakiet A29 pakietem A29PP i dodając nowy pakiet A32PP. Ponadto niezbędne były przeróbki innych pakietów, przeszycie pamięci stałej i wykonanie dodatkowych połączeń na płaterze. Dla kontroli poprawności działania zmodyfikowanego układu przeadresacji opracowano trzy proste autonomiczne testy programowe.

Poza przebudową przeadresatora dokonano również modyfikacji umożliwiającej, za pomocą zredefiniowanego rozkazu o kodzie 125 (MODE), zmianę trybu adresowania oraz trybu skoków z poziomu systemu użytkowego. Uzyskanie tej właściwości wymagało uzupełnienia mikroprogramu realizującego akcję przerwania o operacje składowania stanu przerzutników ESM i EJM. Odtwarzanie stanu obliczeń dokonywane jest następnie przez system operacyjny za pomocą zredefiniowanego rozkazu 172E. Pracę procesora w dotychczasowym standardzie umożliwia wprowadzenie dodatkowego przerzutnika RM. Przy RM=0 obowiązuje podstawowy standard ODRY 1305, a przy RM=1 dozwolone są zmiany stanu przerzutników ESM i EJM za pomocą rozkazu MODE. Stan rejestru RM jest określony przez system operacyjny za pomocą rozkazu 172E, wykonywanego w trakcie realizacji odpowiedniego ekstrakodu wydanego przez system użytkowy.

Wprowadzenie przedstawionych tu modyfikacji wymagało także dokonania zmian w programie zarządzającym Executive E6RM. Zmiany te mają postać modyfikacji wprowadzanych w chwili ładowania programu E6RM do pamięci operacyjnej. Dotyczą one głównie uzupełnienia informacji składowanych w bloku opisu programu użytkowego PIB, redefinicji programu przydziału czasu BATM i wprowadzenia definicji nowych ekstrakodów. Dokonane zmiany powodują jedynie rozszerzenie możliwości programu Executive E6RM, dlatego całe dotychczasowe oprogramowanie działa pod nadzorem tego programu w sposób zgodny z podstawowym standardem systemu ODRA 1305 + Executive E6RM.

Przedstawionych modyfikacji procesora centralnego komputera ODRA 1305 dokonano w Środowiskowym Ośrodku Informatyki Politechniki Poznańskiej w roku 1977. Od tego czasu nie odnotowano żadnego uszkodzenia dodatkowo wprowadzonych elementów logicznych (pomimo „ręcznego" wykonania pakietów), jak również żadnego pośredniego wpływu dokonanych modyfikacji na działanie standardowych programów tej maszyny (w tym również na działanie systemu operacyjnego GEORGE 3). Szczegółowa instrukcja wykonania omawianych modyfikacji zawarta została w raporcie [3].

LITERATURA

[1] Dokumentacja techniczna i programowa maszyny cyfrowej ODRA 1305. ELWRO, Wrocław

[2] Liszyński Z.: System operacyjny dla wieloprocesowych, jednodziedzinowych systemów informatycznych. Materiały Konferencji Naukowo-Technicznej „Informatyka w Wielkopolsce", Poznań, 1978, str. 83—92

[3] Liszyński Z., Waśkowiak A.: Rozwinięte metody adresowania pamięci. Modyfikacje procesora ODRA 1305. Raport nr 53 Środowiskowego Ośrodka Informatyki Politechniki Poznańskiej, Poznań, 1978

[4] Liszyński Z., Waśkowiak A.: Sposób adresacji pamięci operacyjnej w maszynach cyfrowych z rejestrami granicznymi. Patent PRL nr 114989

[5] Opis funkcjonalny maszyny cyfrowej ODRA 1305. Ośrodek Szkolenia MERA—ELWRO—SERVICE, Wrocław, 1974

[6] Ślusarek M.: Realizacja programów współbieżnych w systemie ODRA 1305, INFORMATYKA nr 3/1981, str. 4—6.

JERZY GRABOWSKI
Warszawa

Przegląd mikroprocesorów 16-bitowych (5)

LSI-11

Historię mikrokomputerów wykorzystujących układy LSI-11 warto zacząć od przedstawienia poprzedników rodziny, czyli minikomputerów z serii PDP-11. Komputery te wprowadzone na rynek przez firmę DIGITAL EQUIPMENT CORPORATION w roku 1970 mają 16-bitowe słowo, 16-bitowe rejestry uniwersalne i licznik rozkazów oraz bogaty zbiór operacji maszynowych. Procesor komunikuje się z pamięcią i wszystkimi urządzeniami zewnętrznymi przez 72 linie (56 dla sygnałów, 2 dla zasilania i 14 uziemień) magistrali UNIBUS. Procesor nie ma specjalizowanych rozkazów wejścia-wyjścia, gdyż dostęp do rejestrów urządzeń zewnętrznych odbywa się za pomocą rozkazów przesyłania, a adresy tych rejestrów są odwzorowane w 16-bitowy obszar adresowy pamięci (4096 największych adresów). W komunikacji przez UNIBUS stosuje się złożony protokół wymiany obustronnych potwierdzeń (ang. handshaking), dzięki czemu możliwe jest łatwe dołączenie zarówno szybkich, jak i wolnych urządzeń zewnętrznych.

Pierwszy minikomputer PDP-11/05 zapoczątkował drugą serię maszyn, które są produkowane do tej pory. Dwa spośród jego ośmiu 16-bitowych rejestrów spełniają funkcje licznika rozkazów i wskaźnika stosu systemowego z jednoczesnym zachowaniem wszystkich pozostałych właściwości. Rejestry uniwersalne mogą być używane do operacji arytmetycznych i logicznych, jako rejestry adresowe lub indeksowe. Duży i starannie dobrany zbiór rozkazów, osiem różnych trybów adresowania, 16-bitowy adres pamięci — oto zbiór cech określających minikomputer, o którym można mówić, że otworzył nowy rozdział w dziedzinie możliwości i zastosowań maszyn tej klasy.

Następne minikomputery z tej serii były kolejno modyfikowane:

- rozszerzono zbiór rozkazów arytmetycznych i logicznych, dodając rozkazy mnożenia i dzielenia stałoprzecinkowego, przesunięcia itp. (PDP-11/20)
- wprowadzono adresowanie pamięci operacyjnej powyżej 32 K słów 16-bitowych i wyodrębniono dwa tryby pracy procesora — tryb programu użytkownika i programu jądra systemu z powieleniem rejestru wskaźnika stosu — oraz dodano podstawowe rozkazy operacji zmiennoprzecinkowych (PDP-11/40)
- dodano oddzielny procesor zmiennoprzecinkowy z możliwością równoległego wykonywania operacji z procesorem głównym i bardzo bogatym zbiorem rozkazów działających na sześciu specjalizowanych rejestrach 64-bitowych (PDP-11/45)
- podwojono zbiór rejestrów uniwersalnych dla trybu pracy systemowej
- rozszerzono adres do 22 bitów (PDP-11/70) i wprowadzono szybką, zapełnianą sprzętowo, buforową pamięć półprzewodnikową do przechowywania tych 1024 komórek pamięci, do których były ostatnio odwołania z programu (ang. cache memory); rozwiązanie to służy skróceniu czasu dostępu do rozkazów często wykonywanych (np. w pętli) i zmiennych używanych przez te rozkazy; pamięć ta jest zapełniana asocjacyjnie, bez ingerencji programisty, na podstawie historii wcześniejszych odwołań.

Inne udoskonalenia szły w stronę unowocześniania technologii, dołączania szybszych, półprzewodnikowych pamięci i zmniejszania rozmiarów systemu.

W 1975 roku pojawiły się mikrokomputery zrealizowane w technologii LSI (PDP-11/03), w których zastosowano mikroprocesory LSI-11 lub LSI-11/2 z listą rozkazów PDP-11/35 lub 11/40. Od roku 1979 dostępne są mikroprocesory LSI-11/23 z możliwością adresowania rozszerzonej pamięci (adres 18-bitowy), rozbudowanym zbiorem rozkazów oraz procesorem zmiennoprzecinkowym. Obecnie wprowadza się mikrokomputery będące odpowiednikami PDP-11/70. Wszystkie te procesory są wieloukładowe.

Mikrokomputery LSI-11, zrealizowane w technologii o dużym stopniu scalenia, mają zmodyfikowaną magistralę komunikacyjną, zastępującą UNIBUS. Nowa magistrala, zwana Q-BUS, składa się z 72 linii (14 dla sygnałów, 14 przeznaczonych dla przyszłych zastosowań, 10 linii zasilania i 8 uziemień).

Podstawową cechą zbioru rozkazów PDP-11 i LSI-11 jest pełna symetria funkcji rejestrów uniwersalnych i ogólność trybów adresowania. Cechy te są konsekwencją realizacji sprzętowej, w której działaniem procesora steruje mikroprogram. Skuteczności tej listy rozkazów najlepiej dowodzi bardzo długie, bo kilkunastoletnie, utrzymywanie się maszyn tej serii na rynku komputerowym, a także — podjęcie produkcji ich odpowiedników w krajach RWPG. System MERA-60 produkowany w Polsce wykorzystuje radzieckie odpowiedniki układów LSI-11.

CHARAKTERYSTYKA OGÓLNA

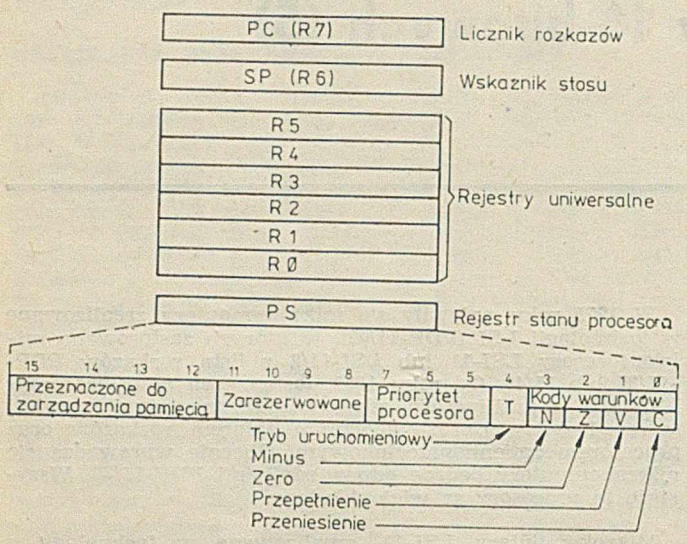
Procesory LSI-11 są wieloukładowe i mikroprogramowane. Składają się z czterech lub pięciu układów pełniących funkcje kontroli, pamięci danych i stałej pamięci mikrokodu (dwa układy) i ewentualnie dodatkowych — dla rozszerzonej arytmetyki stało- i zmiennoprzecinkowej. Procesor LSI-11/23 składa się z: trzech lub czterech układów kontroli, pamięci danych, zarządzania pamięcią wraz z rejestrami procesora operacji zmiennoprzecinkowych, dodatkowego układu pamięci ROM do przechowywania mikrokodu rozkazów zmiennoprzecinkowych w pełnym zestawie rozkazów jak dla PDP-11/45. Mikrokomputer jest wykonywany bez konsoli, z przełącznikami do ręcznych operacji inicjalizacyjnych (tzw. restart, inicjowanie pamięci itd.), ale ma stały program emulacji konsoli przez interakcyjne polecenia z terminala.

Rozkazy mikroprocesora LSI-11 działają na sześciu rejestrach uniwersalnych oraz — liczniku rozkazów PC i wskaźniku zapełnienia stosu systemowego SP. W podstawowym zbiorze rozkazów program ma dostęp do danych 16- lub 8-bitowych oraz do poszczególnych bitów, które można testować i modyfikować. Rozszerzony zestaw rozkazów umożliwia bezpośrednie przetwarzanie danych 32-bitowych. Nowsze układy LSI-11/23 zawierają mikrokod obsługi dostępu i ochrony rozszerzonej pamięci oraz działań na 64-bitowych liczbach zmiennoprzecinkowych podwójnej precyzji.

FORMAT ROZKAZÓW I TRYBY ADRESOWANIA

Na rysunku 1 przedstawiono osiem 16-bitowych rejestrów i 16-bitowe słowo stanu PS procesora LSI-11. Słowo stanu, oprócz wyniku ostatniej operacji (bity C, N, V, Z), określa priorytet procesora, którego wartość definiuje po-

ziom maskowania przerwań (w LSI-11 i LSI-11/2 wykorzystuje się jednak tylko jeden z trzech bitów priorytetu). Słowo PS zawiera także cztery bity przeznaczone do pamiętania bieżącego i poprzedniego trybu pracy procesora, wykorzystywane przez blok zarządzania pamięcią (tylko dla LSI-11/23).



Rys. 1. Podstawowe rejestry mikroprocesora LSI-11

Rejestry uniwersalne R0-R5 mogą być stosowane jako akumulatory, wskaźniki adresu i rejestry indeksowe. Wskaźnik stosu wyróżnia się dodatkowo tym, że jest automatycznie modyfikowany przy wywołaniach procedur i przerwaniach wewnętrznych lub zewnętrznych, wskutek zapisu na stos adresu powrotu lub — dla przerwań — także stanu procesora.

Rozkazy jedno- i dwuargumentowe używają ośmiu różnych trybów adresowania, odwołujących się do rejestrów uniwersalnych lub specjalizowanych. Dodatkowe, specjalne tryby adresowania odwołują się do rejestru PC.

Tabela 1. Tryby adresowania mikroprocesora LSI-11

Tryb	Nazwa	Przykłady	
		zapis symboliczny	kod
0	Rejestrowe	INC R1	005201
1	Pośrednie	INC (R1)	005211
2	Pośrednie z postinkrementacją	INC (R1)+	005221
3	Podwójne pośrednie z postinkrementacją	INC @(R1)+	005231
4	Pośrednie z predekrementacją	INC -(R1)	005241
5	Podwójne pośrednie z predekrementacją	INC @-(R1)	005251
6	Indeksowe	INC 74(R1)	005261
7	Indeksowe pośrednie	INC @200 (R1)	005271 000200

Nawiasy oznaczają pierwszy poziom pośredniości adresowania (przez rejestr), a znak @ — drugi poziom pośredniości (przez pamięć)

W tabeli 1 zamieszczono przykłady zapisu trybów adresowania i kody maszynowe rozkazu jednoargumentowego INC (powiększ wskazany argument o 1). Uważny czytelnik łatwo stwierdzi, że rejestr wymieniony w trybie adresowania jest kodowany na trzech najmniej znaczących bitach rozkazu, a tryb adresowania — na trzech następnych. Pole określające adres w kodzie rozkazu jest więc 6-bitowe, a w trybach 6 i 7 występuje dodatkowe 16-bitowe słowo przemieszczenia. Tryby adresowania 0, 1, 2 i 4 są znane z wcześniejszego opisu innych mikroprocesorów i dotyczą adresowania rejestru (tryb 0) i pamięci przez rejestr: bez autoinkrementacji tryb 1), z predekrementacją (tryb 2) lub postinkrementacją (tryb 4). Tryby adresowania 3 i 5 odpowiadają trybom 2 i 4, zmodyfikowanym o następny poziom pośredniości. Tryb 6 jest klasycznym trybem indeksowym. W opisie trybu 7 trzeba określić kolejność kroków przy obliczaniu adresu efektywnego. Zawartość rejestru jest dodawana do przemieszczenia w przykładzie z tabeli — 200) z drugiego słowa rozkazu, a suma określa adres komórki zawierającej adres argumentu. Tak, więc gdy R1 = = 000100, a 16-bitowa komórka o adresie 300 zawiera liczbę 700, to rozkaz INC 200(R1) spowoduje dodanie 1 do zawartości komórki o adresie 000700 (liczby są tu zapisane ósemkowo). W podobny sposób kodowane są rozkazy dwuargumentowe, używające następnych sześciu bitów na zapis drugiego argumentu.

Wszystkie rozkazy procesora LSI-11 są zapisywane w parzystej liczbie bajtów (1, 2 lub 3 słowa 16-bitowe). W konsekwencji pełnej symetrii trybów adresowania względem obu argumentów, dane i wynik operacji przesyła się bezpośrednio do lub z pamięci, nie używając rejestru na pobranie argumentu lub przechowanie wyniku.

Omawiając tryby adresowania pamięci procesora LSI-11 warto zastanowić się nad skutkiem użycia w rozkazie rejestru PC (R7). Pamiętając, że zachowuje się on analogicznie jak inne rejestry, uzyskamy nowe interpretacje trybów 2, 3, 6 i 7. W tabeli 2 podano te tryby adresowania i specjalny zapis stosowany przy ich kodowaniu w języku symbolicznym.

Tabela 2. Tryby adresowania przez licznik rozkazów

Tryb	Nazwa	Przykłady	
		zapis symboliczny	kod
2	Proste (dane bezpośrednie)	ADD #34, R1	002701 000034
3	Bezwzględne (adres bezpośredni)	ADD @#34, R1	063701 000034
6	Względne	ADD A, R1	066701 (Przemieszczenie)
7	Względne pośrednie	ADD @ A, R1	067701 (Przemieszczenie)

OMÓWIENIE LISTY ROZKAZÓW

Lista rozkazów mikroprocesora LSI-11 jest w Polsce dość dobrze znana ze względu na jej zgodność ze zbiorem rozkazów komputerów serii SM. Dlatego też omówimy jedynie wybrane rodzaje rozkazów (tab. 3).

Niektóre rozkazy działają zarówno na słowach, jak i bajtach. W tym drugim przypadku ich nazwa symboliczna kończy się literą B. Mikroprocesor LSI-11 nie ma specjalizowanych rozkazów wejścia-wyjścia, a ich funkcje spełniają rozkazy przesyłania MOV i MOVB. Nie ma także rozkazów wyrównania dziesiętnego i rozkazów operowania blokami danych. Istnieje natomiast wiele rozkazów arytmetyki dwójkowej, szczególnie dla procesora LSI-11/23, który ma rozkazy operacji zmiennoprzecinkowych działające na liczbach 64-bitowych i rozkazy stałoprzecinkowej arytmetyki 32-bitowej. Można także różnymi sposobami generować przerwania wewnętrzne, korzystając z oddzielnych wektorów przerwań. Ponadto istnieje wiele rozkazów skoków warunkowych i zwykłych, a także — rozkazy kontroli układów zarządzania pamięcią i magistralą mikrokomputera.

W mikroprocesorach rodziny LSI-11 występuje kilka rozkazów szczególnych, które warto krótko omówić. Rozkazy BIT (badanie bitów) i BIS (ustawianie bitów) są typowe dla mikroprocesorów, jednakże rozkaz BIC (kasowanie

bitów) wykonuje operacje maskowania bitów zastępując nieistniejący rozkaz iloczynu logicznego. Rozkaz SOB (odejmowanie 1 i skok, gdy wynik nie jest zerem) umożliwia jednorozkazową kontrolę zakończenia pętli. Rozkaz SXT rozszerza znak bajtu i — podobnie jak ADC (dodanie przeniesienia) — jest wykorzystywany do tworzenia działań wielokrotnej precyzji.

Tabela 3. Przykładowe rozkazy mikroprocesora LSI-11

Zapis symboliczny	Działanie
ROZKAZY JEDNOARGUMENTOWE	
Wielokrotnej precyzji	
ADC(B)	Dodanie przeniesienia
SBC(B)	Odjęcie przeniesienia
SXT	Rozszerzenie znaku
Zmiennopozycyjne (tylko dla LSI-11/23)	
SETF	Ustawienie trybu pojedynczej precyzji
SETD	Ustawienie trybu podwójnej precyzji
SETI	Ustawienie trybu pojedynczej precyzji dla arytmetyki stałoprzecinkowej
SETL	Ustawienie trybu podwójnej precyzji dla arytmetyki stałoprzecinkowej
ROZKAZY DWUARGUMENTOWE	
Logiczne	
BIT(B)	Badanie bitów
BIC(B)	Kasowanie bitów
BIS(B)	Ustawianie bitów (suma logiczna)
XOR	Różnica symetryczna
Zmiennoprzecinkowe (dla LSI-11 i LSI-11/2)	
FADD	Dodawanie
FSUB	Odejmowanie
FMUL	Mnożenie
FDIV	Dzielenie
Zmiennoprzecinkowe (dla LSI-11/23)	
LDF(LDD)	Pobieranie liczby pojedynczej (podwójnej) precyzji
STF(STD)	Zapamiętanie liczby pojedynczej (podwójnej) precyzji
ADDF(ADDD)	Dodanie liczby pojedynczej (podwójnej) precyzji
SUBF(SUBD)	Odjęcie liczby pojedynczej (podwójnej) precyzji
MULF(MULD)	Pomnożenie liczby pojedynczej (podwójnej) precyzji
DIVF(DIVD)	Podzielenie liczby pojedynczej (podwójnej) precyzji
POZOSTAŁE ROZKAZY	
Skoki zwykłe i wywołania procedury	
JMP	Skok
JSR	Skok do procedury
RTS	Powrót z procedury
MARK	Pomocniczy dla przekazu parametrów
SOB	Odjęcie 1 i skok, gdy nie zero
Przerwania wewnętrzne programowe	
EMT	Zgłoszenie emulacji
TRAP	Zgłoszenie użytkowe
BPT	Zgłoszenie wstrzymania
RTI	Powrót z przerwania
RTT	Powrót ze zgłoszenia
Kontrola układu MMU (tylko dla LSI-11/23)	
MFPD(MFPI)	Prześlij z uprzedniego obszaru danych (kodu)
MTPD(DTPD)	Prześlij do uprzedniego obszaru danych (kodu)

Urozmaicony jest zbiór rozkazów warunkowych, w których, podobnie jak dla rozkazu SOB, stosuje się tylko adresowanie względne. Ośmiobitowe przemieszczenie umożliwia zmianę zawartości licznika rozkazów od —127 do +128 słów względem bieżącej wartości PC.

Rozkazy wywołania i powrotu z procedury są dosyć szczególne. Rozkaz wywołania (JSR) ma dwa argumenty, z ktorych jeden musi być rejestrem. Stan tego rejestru, zwanego łączącym (ang. link register), jest przesyłany na stos przed wykonaniem skoku do adresu efektywnego z drugiego argumentu. Przed skokiem i po zapisaniu na stosie bieżącej zawartości, do rejestru łączącego jest wpisywany adres powrotu. Zatem, adres powrotu jest przesyłany na stos tylko wtedy, gdy rejestrem łączącym jest PC. W innym przypadku, adres ten jest zapamiętywany w rejestrze. Rozkaz powrotu (RTS), który odwołuje się do rejestru określonego w poprzednim wywołaniu, przesyła do PC zawartość rejestru i odtwarza poprzedni stan rejestru łączącego, pobierany ze stosu. W przypadku, gdy rejestrem łączącym jest R7 (PC) współdziałanie rozkazów wywołania i powrotu z procedury sprowadza się do przechowania i odtworzenia stanu PC przez stos. Możliwość stosowania rejestru łączącego, zawierając adres rozkazu następującego po wywołaniu, ułatwia przekazywanie parametrów aktualnych procedury, umieszczonych po rozkazie wywołania. Warto dodać, że przy określaniu adresu dla prostego skoku i wywołania procedury, można używać wszystkich trybów adresowania oprócz trybu 0 (nie wolno wykonać skoku do rejestru).

ROZKAZY WEJŚCIA-WYJŚCIA I PRZERWANIA

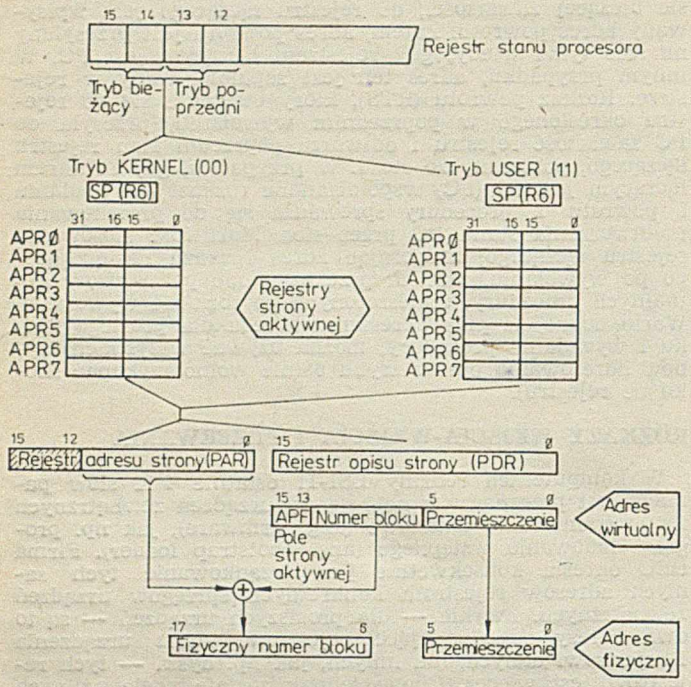
W komputerach rodziny LSI-11 ostatnie 4 K słów pamięci jest przeznaczone na rejestry urządzeń zewnętrznych i oprogramowanie układowe (ang. firmware), jak np. program ładowania wstępnego (ang. bootstrap loader). Firma DEC określa konsekwetnie przyporządkowanie tych samych adresów rejestrom konkretnych sprzęgów urządzeń zewnętrznych. Zwykle — dla prostszych urządzeń — są to dwa adresy odpowiadające rejestrowi stanu urządzenia i rejestrowi danych, dla innych, jak np. dyski — tych rejestrów jest więcej. Ciągle jednak zostaje wiele wolnych adresów do dołączenia nietypowych urządzeń zewnętrznych. Z kolei, stałe rozmieszczenie bitów w obrębie rejestru stanu urządzenia ułatwia tworzenie oprogramowania. Urządzenia mogą być obsługiwane programowo w trybie zapytań (ang. polling), przez obsługę przerwań, lub w trybie bezpośredniego dostępu do pamięci.

Zgłoszenie przerwania powoduje zapisanie na stosie aktualnych zawartości licznika rozkazów PC i rejestru stanu PS oraz wpisanie nowych wartości do PC i PS z dwóch komórek pamięci, przeznaczonych na wektor przerwania. Rozkazy RTI i RTT (powrót z przerwania zewnętrznego i wewnętrznego) odtwarzają poprzedni stan PS i PC, pobierając dwa słowa z wierzchołka stosu. Przerwania maskowalne mogą być blokowane przez określenie 3-bitowego priorytetu procesora (w słowie stanu PS) i porównanie z priorytetem urządzenia. Przerwania niemaskowalne są związane ze szczególnymi sytuacjami, jak np. zanik zasilania czy rozkaz nielegalny. Przerwania wewnętrzne mogą być wywołane przez różne rozkazy i są używane np. do komunikacji z systemem operacyjnym. Brak jest przerwań wewnętrznych związanych z przepełnieniem, powstałym wskutek stałoprzecinkowej operacji arytmetycznej. Z drugiej strony — mikroprocesor LSI-11/23, zawierający układy operacji zmiennoprzecinkowych, generuje wiele przerwań związanych z błędami arytmetyki.

BLOK ZARZĄDZANIA PAMIĘCIĄ MIKROPROCESORA LSI-11/23

Działanie układów zarządzania pamięcią jest warte szerszego omówienia, gdyż stosowane tu rozwiązania można uznać za wzorcowe. Mikroprocesor LSI-11 ma 16 linii adresowych, z czego wynika ograniczenie możliwości adresowania do 64 bajtów pamięci. Natomiast, w mikroprocesorze LSI-11/23 zastosowano 18 linii adresowych, czemu odpowiada fizyczny obszar adresowy o pojemności 256 K bajtów. Odwzorowanie 16-bitowego adresu wirtualnego, generowanego przez program w 18-bitowy adres fizyczny jest realizowane przez jeden układ MOS/LSI. Funkcje tego układu odpowiadają programowym właściwościom procesorów PDP-11/34, 11/40, 11/45. Istnieją także układy LSI-11 realizujące transformację w 22-bitowy obszar adresowy, czyli odpowiadające komputerowi PDP-11/70 (4 M bajtów pamięci).

Dla układu zarządzania pamięcią procesor znajduje się w jednym z dwóch stanów pracy: „program użytkownika" lub „program jądra systemu" (odpowiednio ang. USER i KERNEL). W stanie KERNEL program ma pełną kontrolę nad procesorem, tzn. może wykonywać wszystkie rozkazy. W stanie USER procesor nie może wykonywać operacji zatrzymania programu i działań prowadzących do modyfikacji obszaru innych zadań, których istnienie zakłada się w środowisku wielozadaniowego systemu operacyjnego.



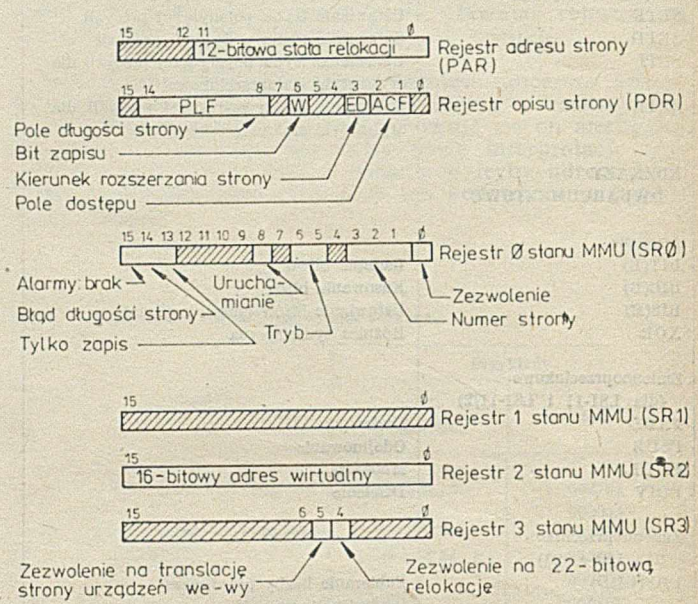
Rys. 2. Tłumaczenie adresu wirtualnego na adres fizyczny przez układ zarządzania pamięcią mikroprocesora LSI-11/23

Na rysunku 2 przedstawiono rejestry układu zarządzania pamięcią (ang. memory management unit MMU) i zilustrowano proces tłumaczenia 16-bitowego adresu wirtualnego na 18-bitowy adres fizyczny. W układzie MMU znajdują się dwie grupy ośmiu 32-bitowych rejestrów zwanych rejestrami strony aktywnej (ang. active page register, APR). Każda grupa jest przypisana jednemu trybowi pracy. Każdy rejestr APR dzieli się na dwie 16-bitowe części zwane rejestrem adresu strony (ang. page address register, PAR) i rejestrem opisu strony (ang. page descriptor register, PDR). Rejestr PAR zawiera 12-bitowy adres strony (16 bitów używa się tylko przy 4 M bajtowym obszarze adresowym), który definiuje adres odniesienia. Rejestr PDR zawiera informację o długości strony, możliwościach jej rozszerzania i określa sposób dostępu (tylko odczyt, zapis, zakaz dostępu).

Gdy układ MMU jest aktywny, to bity 15 i 14 słowa stanu PS (rys. 1) określają, który zbiór rejestrów APR jest używany do tłumaczenia adresu wirtualnego na fizyczny. Stan tych bitów określa dodatkowo wybór odpowiedniego wskaźnika stosu SP (także wtedy, gdy MMU jest odłączony).

Adres wirtualny składa się z trzech części: przemieszczenia wyrażonego w bajtach tzw. DIB, (bity 0—5), numeru bloku (bity 6—12) i pola strony aktywnej (ang. active page field, APF, bity 13—15). Pole APF wyznacza jeden z ośmiu rejestrów APR, który zawiera informację o przemieszczeniu dla adresu fizycznego. Z informacji zawartej w APR, przemieszczenie uzyskuje się dodając 7-bitowy numer bloku wirtualnego do 12-bitowego adresu odniesienia z rejestru PAR. W ten sposób otrzymuje się 12 najbardziej znaczących bitów adresu, określających 12-bitowy numer bloku fizycznego. Mniej znaczące bity (DIB) pobiera się bezpośrednio z adresu wirtualnego (rys. 2). Maksymalna długość fizycznego obszaru adresowanego przez jeden rejestr APR wynosi 8 K bajtów, gdyż można użyć tylko 13 najmniej znaczących bitów 16-bitowego adresu wirtualnego. Długość strony zawarta w PDR może być określona jako wielokrotność 64 bajtów w zakresie od 1 do 128 (od 64 do 8192 bajtów, z krokiem co 64). Oznacza to możliwość dzielenia strony na bardzo małe segmenty o zmiennej długości.

Jednocześnie z generacją adresu fizycznego, zawartość rejestru opisu strony (PDR) jest porównywana z adresem wirtualnym w celu zweryfikowania prawa dostępu. Jeśli zainicjowana operacja dostępu nie jest dozwolona, to układ MMU nie realizuje jej i zgłasza niemaskowalne przerwanie wewnętrzne. Dwa z czterech rejestrów stanu MMU (rys. 3) zawierają informację o przyczynie przerwania. Na rysunku 3 przedstawiono także rozmieszczenie informacji w rejestrach APR i PDR.



Rys. 3. Rejestry bloku zarządzania pamięcią

Złożoność procesu transformacji adresu przez blok zarządzania pamięcią skłania do stwierdzenia, że działanie układu jest dość czasochłonne. Jednak cały proces generacji i akceptacji adresu fizycznego powoduje wydłużenie cyklu pamięci jedynie o 100 ns.

Rozbudowane możliwości ochrony i zarządzania pamięcią powodują, że programowa obsługa układu MMU nie jest łatwa. Wykonuje ją praktycznie zawsze jądro systemu operacyjnego. Firma DEC dopiero niedawno zaczęła dostarczać oprogramowanie podstawowe, które wykorzystuje większość możliwości układu (system operacyjny RSX-11M+). Mimo że kolejne wersje systemu RSX-11M, starannie opracowane i wygodne — z punktu widzenia obsługi systemów wielozadaniowych i wielodostępnych pracujących w trybie nadążnym, były przez wiele lat (od 1976 r.) wykorzystywane przez użytkowników, to system ten nie uwzględniał np. oddzielnej ochrony pamięci dla kodu i danych programu. Ten przykład wskazuje na to, że rozwój oprogramowania często nie nadąża za postępami w dziedzinie technologii układowej.

JACEK OWCZARCZYK
Warszawa

Projektowanie systemu graficznego. Część 2

W systemach graficznych coraz częściej wykorzystuje się konstrukcję modułową. Producenci, oprócz gotowych systemów, oferują także wiele pakietów o określonych funkcjach, co umożliwia zestawienie konfiguracji o parametrach najbardziej dostosowanych do konkretnych wymagań użytkownika. Na rynku komputerowego sprzętu graficznego można zauważyć również specjalizację firm w produkcji poszczególnych podzespołów, takich jak: pakiety z układami systemów graficznych, graficzne urządzenia wejściowe czy monitory. Niektóre firmy, oferujące gotowe systemy graficzne, składają systemy z podzespołów innych producentów, opracowując jedynie własne oprogramowanie. Jednocześnie producenci układów scalonych wprowadzają coraz bardziej zaawansowane monolity (monolityczne układy scalone), które przejmują dotychczasowe funkcje pakietów.

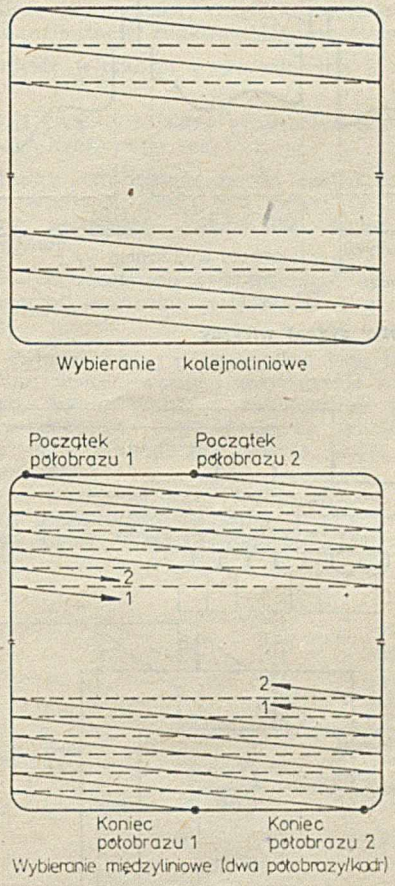
Ta różnorodność środków technicznych stwarza projektantowi systemów graficznych, nie spotykane dotychczas możliwości, ale też zmusza go do podejmowania wielu nowych decyzji. Dotyczą one głównie wyboru technologii w konstrukcji sprzętu; np. wykorzystanie gotowych pakietów upraszcza konstrukcję systemu graficznego, ale narzuca również pewną jego organizację. Niezależnie jednak od technicznej realizacji podstawowych układów rastrowego systemu graficznego, ich funkcje i zasada działania pozostają takie same.

SYGNAŁ WIZYJNY

W monitorze telewizyjnym obraz tworzony jest przez ruch plamki świetlnej o modulowanej luminancji. Niezależnie od treści obrazu ruch plamki odbywa się według pewnego ustalonego schematu wypełniając raster. Bez względu na sposób tworzenia rastra (rys. 1), roboczy ruch plamki (wybieranie) odbywa się od lewej strony do prawej i z góry na dół. W czasie powrotów plamki sygnał luminancji ma wartość większą od poziomu czerni — następuje „wygaszenie", plamka staje się niewidoczna na ekranie. Układy odchylania strumienia elektronów lampy kineskopowej zapewniają prawidłowy bieg plamki, a impulsy synchronizujące — właściwe położenie obrazu względem rastra.

Dostarczany do monitora sygnał wizyjny musi zawierać wszystkie informacje niezbędne do wytworzenia obrazu na ekranie. W przypadku monitorów monochromatycznych, całkowity sygnał wizyjny składa się z trzech sygnałów: sygnału luminancji, impulsów synchronizujących i impulsów wygaszania (rys. 2). Standardy telewizyjne określają zależności czasowe sygnału wizyjnego; dla przykładu, zależności te dla impulsu synchronizacji poziomej zostały przedstawione na rysunku 2. Impuls synchronizacji pionowej jest dość złożony — składa się z serii impulsów wyrównawczych występujących przed i po właściwym impulsie synchronizacji pionowej, który ponadto jest „poprzerywany". Ma to na celu utrzymanie synchronizacji poziomej w czasie pionowego powrotu plamki. Sygnał wizyjny może mieć polaryzację negatywną (wówczas większej jaskrawości odpowiada mniejsza wartość sygnału luminancji) lub pozytywową.

W monitorach kolorowych sygnał wizyjny może być zakodowany według używanego systemu telewizji kolorowej (SECAM, PAL, NTSC) lub też może być dostarczany w postaci trzech sygnałów: R („czerwony" sygnał wizyjny), G („zielony" sygnał wizyjny), B („niebieski" sygnał wizyjny) oraz sygnału synchronizacji i wygaszania.

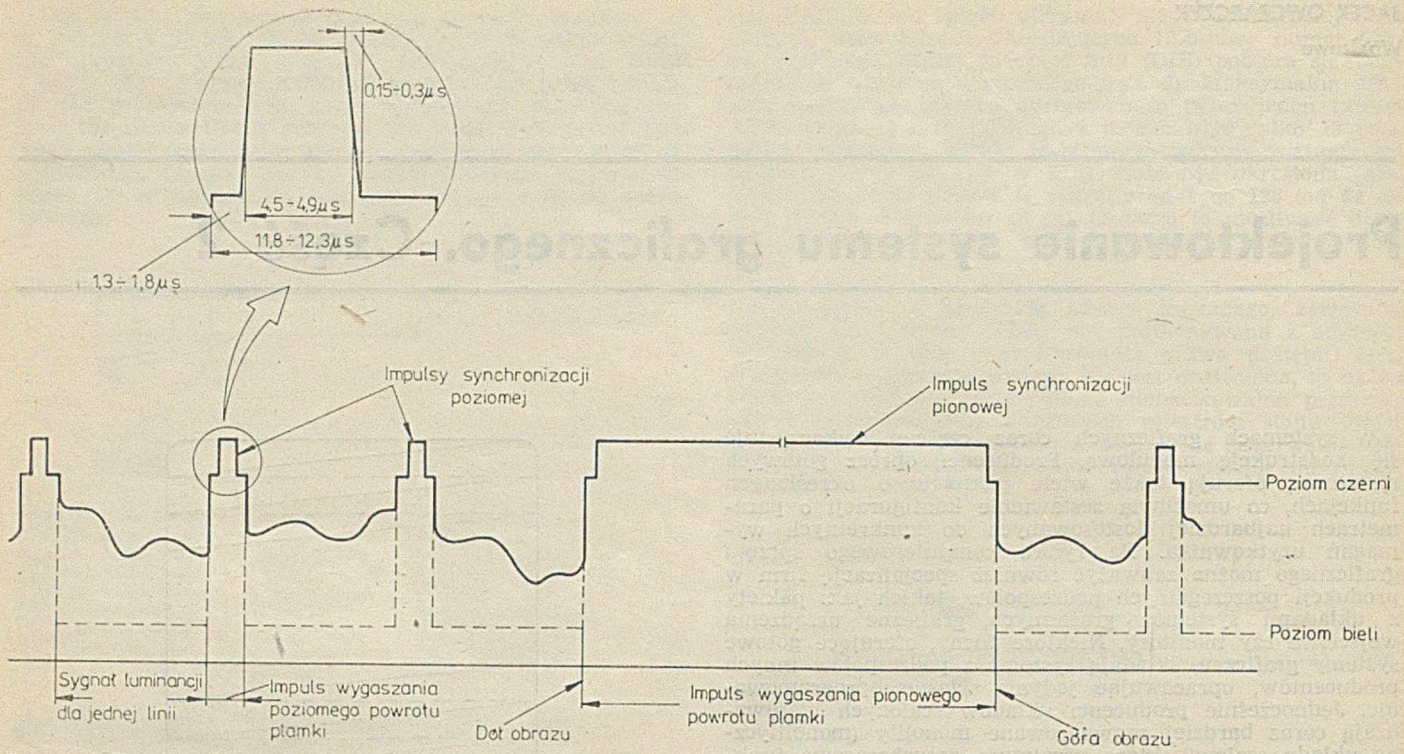


Rys. 1. Sposoby tworzenia rastra

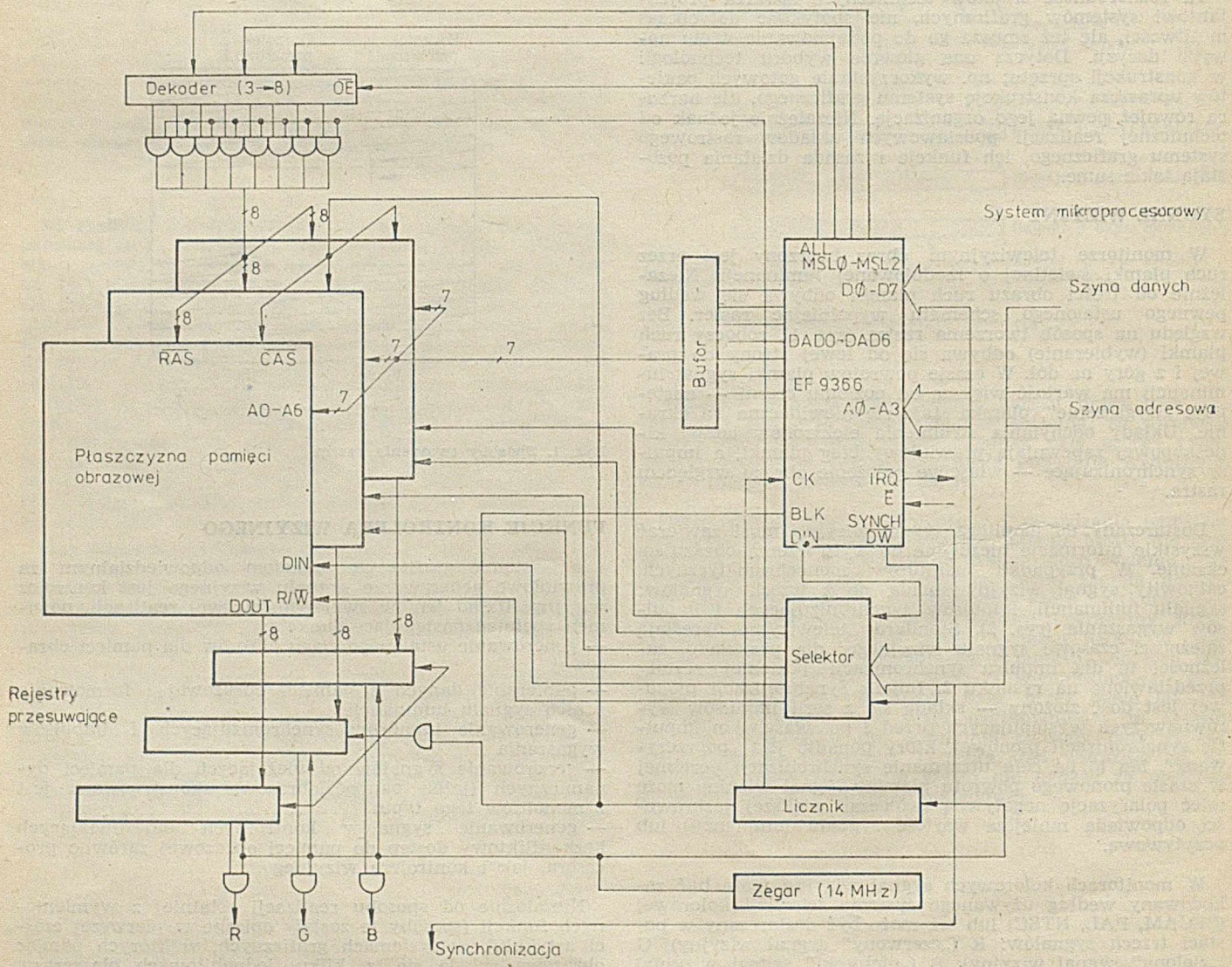
FUNKCJE KONTROLERA WIZYJNEGO

W systemie graficznym układem odpowiedzialnym za prawidłowe generowanie sygnału wizyjnego jest kontroler wizyjny. Układ ten, w swej podstawowej realizacji, powinien spełniać następujące funkcje:
— generowanie ustalonego ciągu adresów dla pamięci obrazowej
— pobieranie danych z pamięci obrazowej i formowanie z nich sygnału luminancji
— generowanie impulsów synchronizujących i impulsów wygaszania
— generowanie sygnałów odświeżających dla pamięci dynamicznych (jeżeli pamięć obrazowa skonstruowana jest z monolitów tego typu)
— generowanie sygnałów kontrolnych umożliwiających bezkonfliktowy dostęp do pamięci obrazowej zarówno procesora, jak i kontrolera wizyjnego.

Niezależnie od sposobu realizacji ostatniej z wymienionych funkcji (sposoby te zostały opisane w pierwszej części artykułu), w systemach graficznych, w których pamięć obrazowa składa się z kilku jednobitowych płaszczyzn, przesyłanie danych z pamięci obrazowej do rejestrów kon-



Rys. 2. Całkowity sygnał wizyjny



Rys. 3. Przykładowe wykorzystanie kontrolera EF9366

trolera wizyjnego wymaga innej szyny danych niż ta, która służy do wymiany informacji pomiędzy procesorem a pamięcią obrazową. W pierwszym przypadku, jednocześnie z każdej płaszczyzny, pobierana jest ustalona grupa bitów (bajt, słowo) i przesyłana do rejestru przesuwającego (a więc rejestrów przesuwających jest tyle, ile płaszczyzn pamięci obrazowej); w drugim — z rejestru danych do każdej płaszczyzny przesyłany (pobierany) jest tylko jeden bit. A zatem przesyłanie danych z procesora do pamięci obrazowej wymaga układów dodatkowych, umożliwiających wybór i modyfikację tylko jednego bitu (cykl: odczyt-modyfikacja-zapis).

Podstawowym elementem kontrolera wizyjnego jest zegar. Z impulsów zegarowych za pomocą dzielników tworzy się cyklicznie adresy komórek pamięci obrazowej i wyprowadza sygnały służące do formowania impulsów synchronizujących i impulsów wygaszania. W konstrukcji kontrolera wizyjnego coraz częściej stosuje się, zamiast układów TTL, specjalizowane układy scalone.

GRAFICZNE KONTROLERY WIZYJNE

Wprowadzone w ostatnim czasie jednostkowe graficzne kontrolery wizyjne w znacznym stopniu upraszczają konstrukcję systemu graficznego. Dwa najbardziej zaawansowane to: EF9365/6 i μPD7220. W porównaniu z dotychczas produkowanymi jednokostkowymi kontrolerami wizyjnymi (jak np. MOTOROLA MC6845), które również mogą być wykorzystane w konstrukcji systemów graficznych — jakkolwiek ich głównym przeznaczeniem są terminale alfanumeryczne, graficzne kontrolery wizyjne spełniają wiele funkcji typowych dla systemów graficznych. Zarówno EF9365/6 jak i μPD7220 przeznaczone są w zasadzie do konstrukcji systemów grafiki komputerowej.

EF9365/6 [1, 2]

Kontroler ten, produkowany przez firmę THOMSON-EFCIS, spełnia następujące funkcje:
— zarządzanie pamięcią obrazową (kontrola zapisu i odczytu, generowanie sygnałów odświeżających dla półprzewodnikowych pamięci dynamicznych)
— formowanie standardowego sygnału wizyjnego (standard CCIR 625 linii); maksymalne wymiary rastra: 512×512 punktów (EF9365), 512×256 punktów (EF9366)
— generowanie znaków alfanumerycznych (z możliwością programowania ich kształtu, wielkości i orientacji)
— generowanie wektorów (z szybkością 1,5 mln punktów/s)
— obsługa pióra świetlnego.

Może on być bezpośrednio przyłączany do typowych systemów mikroprocesorowych z 8-bitową szyną danych, przy czym komunikacja z kontrolerem odbywa się za pomocą 11 rejestrów:

CMD — rejestr rozkazów; możliwe są następujące typy rozkazów: wykreśl wektor, wykreśl znak alfanumeryczny, „wyzeruj" ekran, wykonaj sekwencję pióra świetlnego

CTRL1, CTRL2 — rejestry kontrolne, sterujące dostępem do pamięci obrazowej, przerwaniami, sposobem kreślenia wektorów (linie przerywane, ciągłe, kropkowane); określają one także orientację znaków alfanumerycznych

CSIZE — rejestr wielkości wyświetlanych znaków alfanumerycznych

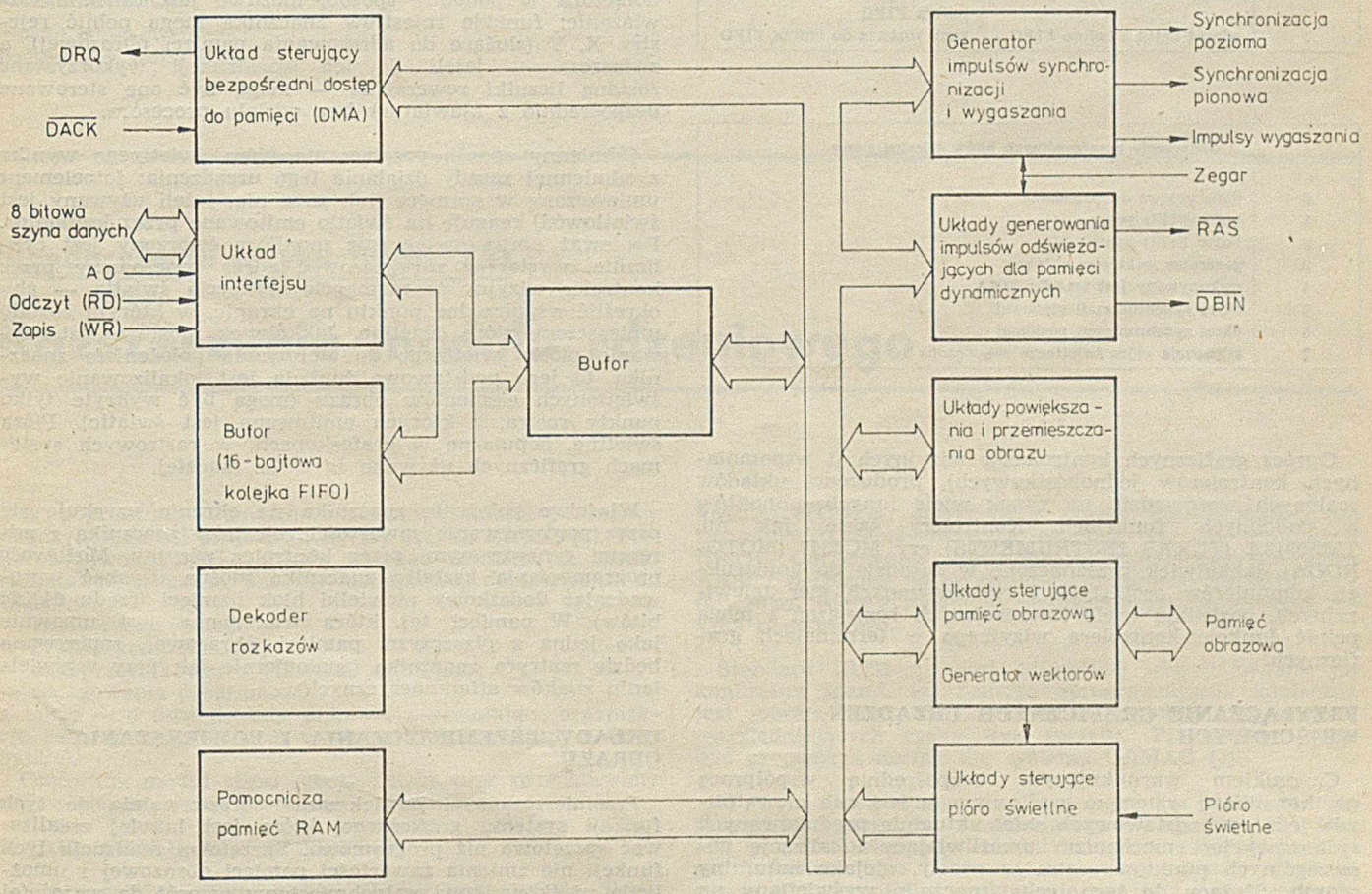
DELTAX, DELTAY — rejestry generatora wektorów; określają przyrosty kreślonego wektora

X, Y — rejestry lokalizujące punkt rastra, który ma być aktualnie wyświetlony; mogą być ustawiane przez generator wektorów, generator znaków lub zewnętrzny system mikroprocesorowy

XLP, YLP — rejestry pióra świetlnego; zawierają adres generowany przez kontroler wizyjny w momencie wykrycia światła

STATUS — rejestr stanu; poszczególne jego bity określają między innymi: wykonywanie i zakończenie sekwencji pióra świetlnego, wykonywanie i zakończenie wykonywania rozkazu, przekroczenie dopuszczalnych wartości w rejestrach X i Y, zainicjowanie przerwania, okres impulsu wygaszania pionowego.

Na rysunku 3 zamieszczono jeden z możliwych (sugerowanych przez firmę THOMSON-EFCIS) sposobów wyko-



Rys. 4. Schemat blokowy kontrolera μPD7220

rzystania kontrolera EF9366 w konstrukcji systemów grafiki komputerowej. Jak widać na rysunku, konstrukcja terminala graficznego (o wymiarach rastra 512×256 punktów, z możliwością wyświetlania ośmiu kolorów) wymaga niewielu dodatkowych elementów.

μPD7220 [3]

Produkowany przez firmę NEC MICROCOMPUTERS INC., graficzny kontroler wizyjny μPD7220 spełnia następujące funkcje:
— zarządzanie pamięcią obrazową 2048×2048×1 (z możliwością zmiany organizacji pamięci obrazowej na np. 1024×1024×4 lub inną)
— generowanie wektorów (z szybkością 1,25 mln punktów/s)
— generowanie standardowych sygnałów wizyjnych lub — przy większych wymiarach rastra — generowanie sygnału wizyjnego wg standardu EIA-RS343 dla rastrów: 1024×792 (wybieranie kolejnoliniowe) i 1024×1024 punktów (wybieranie międzyliniowe)
— generowanie znaków alfanumerycznych
— powiększanie i przemieszczanie obrazu
— obsługa pióra świetlnego
— generowanie sygnałów odświeżających dla półprzewodnikowych pamięci dynamicznych.

Schemat blokowy kontrolera μPD7220 przedstawiony jest na rysunku 4. Podobnie jak EF9365, jest on przeznaczony do konstrukcji systemów grafiki komputerowej i dostosowany do współpracy z 8-bitową dwukierunkową szyną danych. Dodatkowy bufor (o strukturze kolejki FIFO) umożliwia asynchroniczną komunikację z systemem mikroprocesorowym. Znaczenie poszczególnych bitów rejestru stanu kontrolera zamieszczono w tabeli 1. Kontroler μPD7220 może wykonywać 18 rozkazów (tab. 2).

Tabela 1. Zasady komunikacji z kontrolerem μPD7220

A0	Tryb „odczyt"	Tryb „zapis"
0	odczyt rejestru stanu	zapis parametrów rozkazów do bufora FIFO
1	odczyt bajtu z bufora FIFO	zapis rozkazu do bufora FIFO

Oznaczenie poszczególnych bitów rejestru stanu	
0	dane gotowe do przesłania
1	bufor FIFO pełny
2	bufor FIFO pusty
3	generator wektorów aktywny
4	wykonywany jest transfer DMA
5	okres synchronizacji pionowej
6	okres synchronizacji poziomej
7	sekwencja pióra świetlnego zakończona

Oprócz graficznych kontrolerów wizyjnych (i wspomnianych kontrolerów jednokostkowych), producenci układów scalonych wprowadzili na rynek wiele innych monolitów o podobnych funkcjach. Kontrolery takie, jak np. TMS9918A (TEXAS INSTRUMENTS) czy MC6847 (MOTOROLA), jakkolwiek przeznaczone w zasadzie do konstrukcji komputerów osobistych i rozbudowanych gier telewizyjnych, posiadają wiele interesujących rozwiązań i mogą pełnić funkcje kontrolera wizyjnego w terminalach graficznych.

PRZYŁĄCZANIE GRAFICZNYCH URZĄDZEŃ WEJŚCIOWYCH

Czynnikiem warunkującym bezpośrednią współpracę użytkownika z systemem graficznym (a jest ona niewątpliwie jedną z podstawowych zalet aktualnie produkowanych systemów) jest mechanizm umożliwiający lokalizację poszczególnych punktów rastra w prosty, niejako naturalny sposób. Służący do tego celu znacznik, wyświetlany na ekranie monitora, jest ustawiany w dowolnym położeniu za pomocą graficznych urządzeń wejściowych — takich, jak: manipulatory kulowe czy drążkowe, pióra świetlne, klawiatury funkcyjne.

Tabela 2. Lista rozkazów kontrolera μPD7220

Rozkaz	Opis działania rozkazu
RESET	ustawia kontroler w stanie początkowym i określa wymiary rastra
VSYNC	wybiera synchronizację wewnętrzną lub zewnętrzną
CCHAR	określa wysokość znacznika i znaków alfanumerycznych
START	rozpoczyna proces wybierania
ZOOM	określa stopień powiększenia obrazu
CURS	ustala pozycję znacznika
PRAM	określa początkowy adres pamięci obrazowej w procesie wybierania
PITCH	określa szerokość (w bitach) pamięci obrazowej
WDAT	zapis słowa (bajtu) do pamięci obrazowej
MASK	ustawia zawartość rejestru maskującego
FIGS	ustawia parametry generatora wektorów
FIGD	kreśli żądany ciąg wektorów
GCHAR	kreśli znak alfanumeryczny
RDAT	odczyt słowa (bajtu) z pamięci obrazowej
CURD	odczyt pozycji znacznika
LPRD	odczyt rejestrów pióra świetlnego
DMAR	żądanie DMA (odczyt)
DMAW	żądanie DMA (zapis)

Manipulatory przyłączane są do systemu mikroprocesorowego (zarządzającego systemem graficznym) najczęściej przez sprzęg szeregowy RS232/V24, a modyfikacja rejestrów znacznika odbywa się drogą programową. Popularność tego rozwiązania wynika z prostoty realizacji technicznej; producenci graficznych urządzeń wejściowych wyposażają manipulatory w sprzęg RS232C/V24 i jest on również oferowany przez firmy produkujące systemy mikroprocesorowe w postaci gotowych pakietów.

Jakkolwiek klawiatura funkcyjna (← → ↑ ↓) może być dołączona w podobny sposób, możliwe jest tu inne rozwiązanie: funkcję rejestrów znacznika mogą pełnić rejestry X, Y (służące do adresowania pamięci obrazowej), a zwłaszcza — jeżeli do ich konstrukcji wykorzystane zostaną liczniki rewersyjne — mogą być one sterowane bezpośrednio z klawiatury bez udziału procesora.

Odmienny sposób przyłączania pióra świetlnego wynika z odmiennej zasady działania tego urządzenia: fotoelement umieszczony w oprawce (lub poza nią, jeżeli używany jest światłowód) reaguje na światło emitowane przez luminofor. Ponieważ obraz na ekranie monitora tworzony jest cyklicznie, wystarczy zarejestrować adres generowany przez kontroler wizyjny w momencie wykrycia światła — aby określić współrzędne punktu na ekranie, w którym zostało umieszczone pióro świetlne. Jakkolwiek możliwe jest używanie pióra świetlnego do sterowania położeniem znacznika, to jego podstawową funkcją jest lokalizowanie wyświetlonych elementów obrazu (mogą być wykryte tylko punkty rastra, z których emitowane jest światło). Pióra świetlne, popularne w grafoskopach, w rastrowych systemach graficznych używane są coraz rzadziej.

Właściwe położenie znacznika na ekranie uzyskuje się przez porównywanie zawartości rejestrów znacznika z adresami generowanymi przez kontroler wizyjny. Możliwość programowania kształtu znacznika można uzyskać wprowadzając dodatkowy niewielki blok pamięci (rzędu 64×64 bitów). W pamięci tej, która traktowana jest umownie jako jedna z płaszczyzn pamięci obrazowej, zapisywana będzie matryca znacznika (analogicznie jak przy wyświetlaniu znaków alfanumerycznych).

UKŁADY PRZEMIESZCZANIA I POWIĘKSZANIA OBRAZU

Przemieszczanie i powiększanie obrazu należą do tych funkcji systemu graficznego, które jest łatwiej zrealizować sprzętowo niż programowo. Sprzętowa realizacja tych funkcji nie zmienia zawartości pamięci obrazowej i umożliwia użytkownikowi natychmiastowy powrót do wcześniej wyświetlanego obrazu.

Przesuwanie obrazu w kierunku pionowym można uzyskać przez wprowadzenie dodatkowych rejestrów, w których zostanie zapisany początkowy adres pamięci obrazowej, generowany przez kontroler wizyjny po impulsie wygaszania pionowego powrotu plamki (początkowy adres w procesie wybierania). Analogiczna modyfikacja adresów umożliwia przesuwanie obrazu w kierunku poziomym. Wyświetlany na ekranie obraz jest zawijany (ang. wrapped around), to znaczy, że przy przesuwaniu obrazu np. do góry, znikająca część pojawia się u dołu.

Efekt powiększenia obrazu uzyskuje się przez wielokrotne powtórzenie kolejnych punktów obrazu zarówno w kierunku poziomym jak i pionowym. Powiększenie nie wnosi tu nowej informacji, pozwala jednak użytkownikowi przeanalizować szczegóły wyświetlanego obrazu w większej skali. Jest ono realizowane przez wprowadzenie układu dzielników i zwolnienie sygnałów zegarowych sterujących pobieraniem przez kontroler wizyjny kolejnych bajtów (słów) z pamięci obrazowej. Dla podstawowej wersji układu powiększanie może być dwu-, cztero- lub ośmiokrotne i dotyczy danych, które mają być wyświetlane w lewym górnym rogu ekranu. W przypadku, gdy w systemie jest również układ przemieszczania obrazu, nie jest to istotne ograniczenie. Aktualizacja rejestrów przemieszczania i powiększania obrazu odbywa się w trakcie impulsu wygaszania pionowego powrotu plamki.

CHARAKTERYSTYKA SPRZĘTU SYSTEMU INSYGRAF

Przedstawione rozważania są wynikiem doświadczeń autora zebranych w trakcie projektowania i konstrukcji systemu przetwarzania obrazów INSYGRAF (**IN**terakcyjny **SY**stem **GRAF**iczny). Jakkolwiek na wybór rozwiązań i parametry systemu miały wpływ, oprócz planowanych zastosowań, również ograniczone możliwości finansowe autora, to organizacja sprzętu umożliwia prostą jego rozbudowę. Dotyczy to zwłaszcza wielkości pamięci obrazowej.

Niektóre parametry systemu INSYGRAF:

procesor: mikrokomputer LSI-11/2 (DEC)
wymiary rastra: 512×256 punktów — dla znaków alfanumerycznych (32 wiersze × 64 znaki); 256×256 punktów — dla informacji graficznej
organizacja pamięci obrazowej: 256×256×4, pamięć obrazowa skonstruowana z monolitów półprzewodnikowej pamięci statycznej HITACHI HM6116P-3 (CMOS, czas dostępu: 150 ns, organizacja: 2K×8 bitów)
kontroler wizyjny wykonany z układów TTL; sprzętowo realizowane — wyświetlanie znaków alfanumerycznych, powiększanie i przemieszczanie obrazu
znacznik, w formie dwóch prezcinających się linii, sterowany bezpośrednio z czteroklawiszowej klawiatury funkcyjnej
dwa sprzęgi typu RS232C/V24: pierwszy przeznaczony jest dla konsoli operatorskiej, drugi (z szybkością transmisji do 38,4 tys. bodów) — do komunikacji z innymi systemami komputerowymi
wyjście wizyjne: YXS1V_{pp}.

* * *

Obie części artykułu obejmują jedynie najważniejsze zagadnienia związane z projektowaniem sprzętu systemu graficznego. Stanowią one odbicie aktualnych tendencji zarówno w organizacji, jak i w technologii rastrowych systemów graficznych, przy czym a k t u a l n e — zwłaszcza w odniesieniu do technologii rastrowych systemów graficznych — oznacza dziś raczej miesiące niż lata.

LITERATURA

[1] Lambinet P., Rey C.: Coprocesseurs graphiques EF 9365/9366 pour terminaux: une solution economique. Minis et Micros, nr 136, 1981
[2] Lambinet P., Rey C.: Coprocesseurs graphiques EF 9365/9366: les modalites d'exploitation. Minis et Micros, nr 138, 1981
[3] Wise J. L., Szejnwald H.: Display controller simplifies design of sophisticated graphics terminals. Electronics, vol. 54, nr 7, 1981.

WACŁAW ISZKOWSKI
MAREK MANIECKI
Instytut Informatyki
Politechnika Warszawska

Standaryzacja języka BASIC (3)

Rozszerzenia modułu rdzeniowego

W poprzednim numerze omówiliśmy moduł rdzeniowy międzynarodowej propozycji normy języka BASIC [1]. Pełne możliwości języka można uzyskać rozszerzając moduł rdzeniowy o cztery moduły dodatkowe, dostosowane do zagadnień:
— czasu rzeczywistego
— grafiki komputerowej
— stałoprzecinkowej arytmetyki dziesiętnej
— redagowania programów,
a także — o dwa moduły plikowe, zapewniające organizację rekordów w formacie wewnętrznym i rodzimym (ang. native).

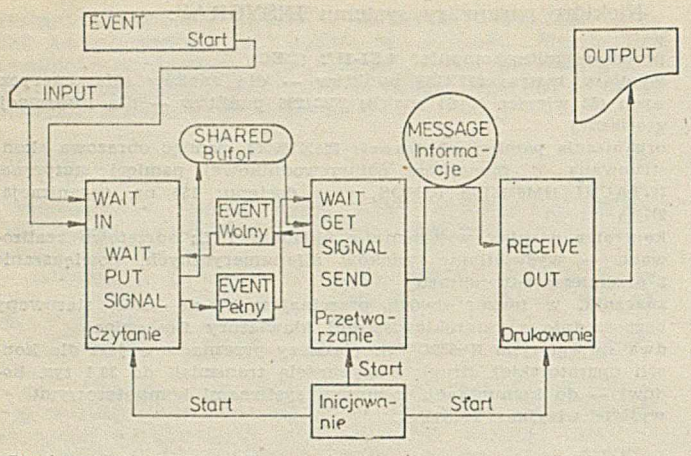
Omówimy moduł czasu rzeczywistego oraz przedstawimy najważniejsze cechy pozostałych modułów.

MODUŁ CZASU RZECZYWISTEGO

Pod auspicjami Europejskiej Grupy Roboczej ds. Przemysłowych Systemów Komputerowych (EWICS) powołano komitet (TC2) do opracowania normy języka BASIC dla zastosowań przemysłowych z uwarunkowaniami czasowymi (IRTB). Opracowane propozycje [2] zostały przyjęte przez ANSI i dołączone do omawianego przez nas projektu normy, jako jeden z modułów rozszerzających możliwości języka rdzeniowego.

Standard IRTB nie jest oczywiście ukierunkowany na konkretny sprzęt. Do pełnego oprogramowania konieczne jest opracowanie specyficznych konstrukcji językowych, uwzględniających cechy tego sprzętu. Przykładem może być propozycja normy dla systemu CAMAC [3].

W celu zilustrowania możliwości języka posłużymy się przykładem programu zawierającego w uproszczonej formie wszystkie elementy składni modułu czasu rzeczywistego. Zadaniem tego programu jest wczytywanie, przetwarzanie, a następnie — drukowanie wartości numerycznych. Strukturę programu przedstawiono na rysunku.



Struktura programu

W programie działają współbieżnie cztery procesy (czynności współbieżne): proces inicjowania, który z kolei uaktywnia trzy następne procesy — czytanie, przetwarzanie i drukowanie. Proces czytania, po aktywacji, oczekuje na zdarzenie zewnętrzne, odwzorowujące przerwanie z czytnika, po czym cyklicznie czyta dane z procesu zewnętrznego (czytnika) i wysyła je do pola wspólnego **(bufor)**, synchronizując współpracę z procesem przetwarzania. Proces przetwarzania, po uaktywnieniu i zsynchronizowaniu swojej pracy z procesem czytania, odbiera dane ze wspólnego pola **bufor** i wysyła je — korzystając z mechanizmu przesyłania wiadomości (ang. message) — do procesu drukowania.

```
 10 !
 20 !
 30 !
 40 STRUCTURE dane: STRING
 50 STRUCTURE bufor_we : STRING, NUMERIC
 60 STRUCTURE wiadomosc : STRING, NUMERIC (5)
 70 STRUCTURE wyniki : 5 OF STRING
 80 !
 90 PROCESS EVENT start "INT_TR"
100 PROCESS INPUT czytnik OF dane "BUF_TR"
110 PROCESS OUTPUT drukarka OF wyniki "BUF_LP11"
120 !
130 SHARED bufor OF bufor_we
140 !
150 MESSAGE informacja OF wiadomosc
160 !
200 !
210 !
300 PARACT inicjowanie
310   WAIT TIME "07:30:00"
320   START czytanie
330   START przetwarzanie
340   START drukowanie
350   PRINT "start of system"
360 END PARACT
400 !
405 PARACT czytanie URGENCY 5
410   CONNECT EVENT start
415   WAIT EVENT start
420   DISCONNECT EVENT start
425 znacznik$:=""
430 DO
435   IN FROM czytnik TO dana$
440   IF  dana$"****"THEN
445     znacznik$:=dana$
450     wartosc := 0
455   ELSE
460     wartosc := VAL (dana$/
465   ENDIF
470   WAIT EVENT wolny
475   PUT TO bufor FROM znacznik$, wartosc
480   SIGNAL pelny
485 LOOP UNTIL znacznik$ = ""
490 PAR STOP
495 END PARACT
500 !
505 PARACT przetwarzanie URGENCY 10
510   DIM tab (5)
515   indeks := 1
520   SIGNAL wolny
525   DO
530     WAIT EVENT pelny
535     GET FROM bufor TO eof$, liczba
540     SIGNAL wolny
545     tab (indeks) :=liczba
550     indeks:=indeks+1
555     IF eof$="****" THEN
560       FOR i:=indeks TO 5
565         tab (i):=0
570       NEXT i
575       indeks := 6
580     ENDIF
585     IF indeks > 5 THEN
590       SEND TO informacja FROM eof$, tab( )
592       indeks := 1
594     ENDIF
596   LOOP
598 END PARACT
600 !
610 PARACT drukowanie URGENCY 20
620   DIM w(5)
630   DO
640     RECEIVE FROM informacja TO, alfa$, w ( )
650     OUT TO drukarka FROM STR$(w(1)), STR$(w(2)),
                        STR$(w(3)), STR$(w(4/) STRS(w(5))
660   LOOP UNTIL alfa$ <>"****"
670   STOP
680 END PARACT
900 END
```

Struktura programu

Program w języku BASIC, rozszerzony o moduł czasu rzeczywistego, jest statycznie podzielony na trzy części:

- sekcja deklaracji struktur do komunikacji ze środowiskiem, w którym program będzie wykonywany (w przykładzie: linie 10—120)
- zbiór czynności (ang. activity) współbieżnych (linie 300—360, 400—495, 500—598, 600—680)
- sekcja procedur zewnętrznych.

Dynamiczne działanie programu polega na współbieżnym wykonaniu sekwencyjnych czynności, które mogą komunikować się ze sobą oraz ze środowiskiem, w którym pracują. Komunikacja może odbywać się przez trzy rodzaje tzw. portów:

- port procesów wejścia-wyjścia — komunikacja ze sprzętem (linie 90, 100, 110)
- port danych dzielonych — komunikacja przez dane wspólne, np. bazy danych (linia 130)
- port wiadomości — komunikacja między czynnościami współbieżnymi (linia 150).

Czynności współbieżne

Czynności współbieżne są definiowane przez instrukcje sekcji równoległych. Początkiem sekcji równoległej jest instrukcja początku sekcji PARACT, a zakończeniem — instrukcja END PARACT. Sekcje równoległe tworzą niezależne jednostki programu, a więc wszystkie użyte w nich identyfikatory obiektów są w nich lokalne. Ponadto instrukcje sterujące mogą odnosić się wyłącznie do linii należących do tej samej sekcji.

W ramach sekcji instrukcje są wykonywane sekwencyjnie, tzn. rozpoczęcie wykonania kolejnej instrukcji następuje po wykonaniu poprzedniej. Natomiast w skali dwóch i więcej sekcji równoległych istnieją takie pary instrukcji, należące do różnych sekcji, dla których kolejność wykonywania nie jest określona. Takie wykonywanie instrukcji sekcji równoległych definiuje czynności współbieżne.

W każdej chwili wykonywania programu czynność współbieżna może być w jednym z następujących stanów:
— aktywnym — po uaktywnieniu czynności
— zatrzymania — po wykonaniu instrukcji zakończenia sekcji lub wystąpienia błędu (ang. fatal error)
— oczekiwania — po zawieszeniu instrukcją czekania lub komunikacji (w oczekiwaniu na określone zdarzenie, upłynięcie określonego czasu lub wymianę wiadomości).

Po uruchomieniu programu, w stanie aktywnym znajduje się czynność określona przez leksykalnie pierwszą sekcję równoległą (w przykładzie jest to sekcja inicjowania

w liniach 300—370). Uaktywnienie kolejnych czynności następuje po wykonaniu instrukcji START. W przykładzie — instrukcja START **czytanie** (linia 320) uaktywnia więc sekcję czytania, po czym obie sekcje wykonują już czynności współbieżne. Sekcja startowania uaktywnia zatem trzy pozostałe sekcje. Trudno wobec tego przewidzieć, kiedy pojawi się napis z linii 350 — może po dość długim okresie działania programu! Próba aktywacji czynności, która nie jest w stanie zatrzymania, powoduje wystąpienie błędu.

Z każdą sekcją równoległą może być statycznie związany priorytet (identyfikowany jako URGENCY), który jest uwzględniany w procesie szeregowania czynności znajdujących się w stanie aktywnym. Przyjmuje się, że im mniejsza wartość priorytetu, tym ważniejsze są czynności danej sekcji.

Zatrzymanie wykonywania czynności w danej sekcji równoległej następuje po wykonaniu instrukcji końca sekcji END PARACT, instrukcji zatrzymania sekcji PARSTOP oraz po wykonaniu w dowolnej sekcji instrukcji zatrzymania STOP lub instrukcji wywoływania następnego programu CHAIN. W naszym przykładzie czynności sekcji startowania zostaną zatrzymane po wykonaniu instrukcji z linii 370 END PARACT, a sekcji czytania — po wykonaniu instrukcji linii 490 PARSTOP; pozostałych (a także obu wymienionych, jeżeli wcześniej nie zostały zatrzymane) — po wykonaniu instrukcji z linii 670 STOP.

Synchronizacja czynności

Do synchronizacji czynności współbieżnych służą instrukcje synchronizacji zdarzeniami oraz instrukcje komunikacji za pomocą wiadomości. Możliwa jest też synchronizacja czynności danej sekcji przez zegar systemowy.

Instrukcje oczekiwania, zależne od zegara systemowego, określają okres wstrzymania czynności danej sekcji równoległej. Instrukcja budzenia WAIT TIME **wyrażenie_napisowe** lub WAIT TIME **wyrażenie_numeryczne** określa chwilę uaktywnienia czynności w zależności od aktualnego czasu. Przykładowo — instrukcja WAIT TIME "07:30:00" w linii 310 ponawia wykonywanie programu o godzinie 7.30. Jeżeli więc program zostanie uruchomiony po godzinie 7.30, to kolejna instrukcja (z linii 320) zostanie wykonana dopiero następnego dnia tuż po godzinie 7.30. Druga postać tej instrukcji podaje czas w sekundach, które upływają od północy, np.:

WAIT TIME (7*60+30)*60

Instrukcja opóźnienia:

WAIT DELAY **wyrażenie_numeryczne**

wstrzymuje działanie danej sekcji równoległej na okres liczby sekund równy wartości wyrażenia.

Synchronizacja zdarzeniami jest prostym mechanizmem realizowanym przez dwie instrukcje:

WAIT EVENT **identyfikator** oczekiwanie na zdarzenie
SIGNAL **identyfikator** sygnalizacja zdarzenia

Każda z tych instrukcji, związanych przez ten sam identyfikator, powinna wystąpić w innej sekcji równoległej.

Przykładowo — w synchronizacji sekcji czytania i przetwarzania zakłada się na wstępie, że zdarzenia (w naszym przypadku identyfikowane jako **wolny** i **pełny**) nie wystąpiły. Tak więc czynności sekcji czytania zostaną wstrzymane po osiągnięciu linii 470, jeżeli wcześniej nie była wykonana instrukcja SIGNAL **wolny** z linii 520, oznaczająca zdarzenie. Zakończenie wykonania instrukcji oczekiwania na zdarzenie powoduje zmianę stanu zdarzenia. Podobna sytuacja występuje dla pary instrukcji identyfikowanej zdarzeniem **pełny**, przy czym instrukcja SIGNAL **pełny** z linii 480 sekcji czytania sygnalizuje zajście zdarzenia dla sekcji przetwarzania w linii 530 WAIT EVENT **pełny**.

Sytuacją błędną, powodującą protest w programie, jest zasygnalizowanie zdarzenia, gdy poprzednie nie zostało jeszcze „odebrane". Zatem instrukcja SIGNAL powinna występować tylko w jednej sekcji, a instrukcje oczekiwania WAIT — w dowolnej liczbie różnych sekcji. Z instrukcją oczekiwania można związać zwrot TIMEOUT **wyrażenie_numeryczne**, oznaczający powstanie protestu, jeżeli zdarzenie nie wystąpi po upływie tylu sekund, ile podaje wartość wyrażenia.

Zajście zdarzenia może być też efektem zewnętrznym względem danego programu, jak — na przykład — przerwanie pracy procesora. Tego rodzaju zdarzenia są sygnalizowane przez procesy zdarzeń o treści zależnej od implementacji. Przykładowo — linia 100 zawiera instrukcję specyfikacji takiego procesu dla zdarzenia **start**. Informacja w postaci napisu jest przeznaczona dla translatora. W sekcji oczekującej na dane przerwanie, wykorzystywana jest instrukcja oczekiwania WAIT EVENT **start** (linia 415). Dodatkowo, analogicznie jak dla procesora rzeczywistego, należy w danej sekcji oczekującej na zdarzenie zewnętrzne dopuścić (odblokować) zgłoszenie takiego zdarzenia (instrukcja CONNECT z linii 410), a potem ewentualnie je zablokować.

Deklaracja struktur komunikacji

Do przesyłania informacji między poszczególnymi sekcjami służą struktury komunikacji. Podstawowym obiektem jest definicja typu strukturalnego, będąca wzorcem przesyłanej informacji, specyfikująca liczbę oraz typ elementu informacji. Rozróżniane są typy numeryczne (NUMERIC) i napisowe (STRING). Dodatkowo możliwe jest zastosowanie tzw. repetytora (n OF) oraz deklaracji tablicy o elementach danego typu. W przykładzie zadeklarowano cztery wzorce o identyfikatorach **dane** (linia 40), **bufor_we** (linia 50), **wiadomosc** (linia 60) i **wyniki** (linia 70). Każdy z wzorców zawiera inny układ elementów.

Oprócz deklaracji struktur, deklarowane są procesy opisujące otoczenie, a więc procesy wprowadzania lub wyprowadzania wartości według przypisanych im struktur. Można też deklarować obszary dzielone, dostępne dla różnych sekcji oraz wiadomości przesyłane między sekcjami. Każdy z tych obiektów jest związany z jedną z zadeklarowanych struktur.

Procesy wejścia i wyjścia

Deklaracja procesów obsługi urządzeń zewnętrznych umożliwia standaryzację zasad współpracy programu z różnymi typami urządzeń w różnych systemach cyfrowych. Przykładowo — w linii 90 zadeklarowano proces wprowadzania (wejścia) o identyfikatorze czytnik i wartościach wprowadzanych według wzorca **dane**. Napis **"BUF_TR"** jest informacją dla translatora wiążącego tę deklarację z rzeczywistym procesem obsługi urządzenia. Analogicznie, w linii 110 zadeklarowano proces wyjścia o identyfikatorze **drukarka** i wartościach wyprowadzonych według wzorca **wyniki** (niekoniecznie musi być to drukarka). Istnieje też trzeci rodzaj procesu tego rodzaju — proces wejścia-wyjścia, deklarowany instrukcją:

PROCESS OUTIN...

Współpraca programu użytkowego z procesami wejścia i wyjścia odbywa się za pomocą instrukcji wejścia IN FROM (np. w linii 435) oraz instrukcji wyjścia OUT TO (np. w linii 650). Ostatnim elementem składni tych instrukcji jest lista znanych wartości do wprowadzenia lub lista wyrażeń, których wartości mają być wyprowadzone. Z instrukcją wejścia można związać omawiany już zwrot TIMEOUT.

Zmienne dzielone

W deklaracji struktur możliwe jest zadeklarowanie obszaru o danej strukturze, który jest dostępny dla każdej z sekcji programu. Obszar ten umożliwia przenoszenie informacji z jednego do wielu procesów. W przykładzie (linia 130) zadeklarowano jeden obszar wspólny o identyfikatorze **bufor** i wzorcu struktury danych **bufor_we**.

Dostęp do tego obszaru jest możliwy przez instrukcje wypełniania PUT TO oraz pobierania GET FROM. Przykładem użycia takiego obszaru jest linia 475 oraz linia 535 w innej sekcji. Oczywiście, należy pamiętać o synchronizacji dostępu do obszaru wspólnego, gdyż w przeciwnym razie może wystąpić błędna sekwencja dostępu (np. PUT, GET, GET, PUT, PUT, PUT, ...). W powyższym przykładzie celowi temu służy synchronizacja zdarzeniami **wolny** i **pełny**.

Przesyłanie wiadomości

Ostatni mechanizm komunikacji, realizowany przez przesyłanie wiadomości, jest podobny w założeniach do mechanizmu procesów komunikacyjnych Hoare'a oraz mechanizmu zastosowanego w języku ADA. W sekcji deklaracji struktur deklaruje się obiekt nazywany wiadomością (MESSAGE) o danym wzorcu struktury. Do komunikacji za pomocą wiadomości służą instrukcje wysłania SEND TO oraz odebrania RECEIVE FROM. Podstawową zasadą ich współdziałania jest takie wzajemne synchronizowanie obu sekcji równoległych, aby w każdej z nich była wykonywana jedna z instrukcji tej pary. Wiadomość jest więc przesyłana bezpośrednio i nie jest nigdzie buforowana.

W przykładzie, po zadeklarowaniu wiadomości w linii 150, metodą tą zsynchronizowano wymianę informacji między sekcjami przetwarzania i drukowania. Odpowiednia para instrukcji wysyłania i odbierania występuje w linii 590 i 640. Dodatkowo należy zwrócić uwagę na konieczność istnienia jednego adresata (odbierającego) wiadomości danego typu oraz możliwość istnienia wielu nadawców (wysyłających) wiadomości tego typu.

Jak można zorientować się z podanego, niepełnego opisu modułu czasu rzeczywistego, powstaje język o bardzo dużych możliwościach, ale też niezbyt łatwy w użytkowaniu. Stosowanie go wymaga dobrej znajomości zasad programowania współbieżnego.

GRAFIKA KOMPUTEROWA

Moduł grafiki komputerowej zawiera mechanizmy tworzenia rysunków i wykresów na różnego rodzaju urządzeniach graficznych. Prostokątny obszar, na którym pojawiają się rysunki, jest nazywany ekranem, przy czym dla każdego rodzaju urządzenia wymiary określa się w implementacji. Można je zmienić (zmniejszyć) instrukcją SET z kluczem BOUNDS, np.:

```
10 SET BOUNDS 1, .66
```

gdzie pierwszy parametr podaje względną szerokość, a drugi względną wysokość ekranu.

Rysunki mogą pojawiać się tylko na części ekranu, tj. na prostokącie określonym czterema współrzędnymi, przy czym można zdefiniować prostokąt problemu (instrukcja SET z kluczem VIEWPORT), w którym tworzone są rysunki oraz — prostokąt ekranu (instrukcja SET z kluczem WINDOW), który określa fizyczne współrzędne na ekranie. Poznanie bieżących wartości definiujących omówione obszary umożliwia instrukcja ASK, np.:

```
100 ASK BOUNDS szer, wys
110 ASK VIEWPORT X1, X2, Y1, Y2
```

Moduł ma również mechanizmy umożliwiające zmianę rodzaju rysowanych linii (np. grubość, kolor), także wybór koloru tła rysunku. Parametry te mogą być ustawione instrukcją SET z odpowiednim kluczem, natomiast instrukcja ASK umożliwia poznanie ich aktualnych wartości, np.:

```
10 SET LINE STYLE 1
20 ASK BACKGROUND COLOR C1
30 SET COLOR RED
```

Do rysowania linii punktów służą instrukcje PLOT i MAT PLOT, analogiczne do instrukcji PRINT. Parametrami instrukcji PLOT są pary wyrażeń numerycznych, których wartości określają współrzędne punktów lub linii do narysowania, np.:

```
100 LET PI=3.14159265
110 PLOT 0,0; 0,1; PI,1; PI,-1; 0,-1; 0,0
110 FOR X = 0 TO PI STEP.01
120 PLOT (X,SIN(X));
130 NEXT X
```

Parametrem instrukcji MAT PLOT powinna być tablica (dwuwymiarowa) lub dwa wektory zawierające współrzędne punktów.

Dodatkowe możliwości zapewniają instrukcje:

- MAT FILL — wypełnienie obszaru kolorem lub rysunkiem
- GRAPHIC PRINT — wyrysowanie napisu z określoną wysokością i orientacją znaków
- GRAPHIC INPUT — wczytanie położenia wskazanego (np. piórem świetlnym) przez operatora urządzenia.

Można również dokonywać przekształcania rysunków (funkcje ROTATE, SHEAR, SHIFT, SCALE), a także tworzyć procedury graficzne, definiowane jak podprogramy (słowo kluczowe PICTURE), a wywoływane instrukcją PLOT.

STAŁOPRZECINKOWA ARYTMETYKA DZIESIĘTNA

Klasyczne obliczenia komputerowe dopuszczały dwójkową arytmetykę całkowitą oraz stało- i zmiennoprzecinkową (ang. integer binary, fixed binary, floating binary). Do obliczania wartości typu rzeczywistego w zmiennoprzecinkowej arytmetyce dwójkowej wprowadzono błąd dokładności oraz podwójną konwersję wartości.

Nowe konstrukcje procesorów, przez rozszerzenie listy rozkazów o operacje na reprezentacji dziesiętnej liczb w kodzie EBCD, stwarzają możliwość zastosowania zmiennoprzecinkowej arytmetyki dziesiętnej. Oczywiście, możliwe jest i tutaj powstawanie błędu dokładności obliczeń, związane z charakterem operacji zmiennoprzecinkowych.

W wielu zastosowaniach, jak na przykład w bankowości, wymaga się dokładności „co do grosza", bez zaokrągleń. Moduł stałoprzecinkowej arytmetyki dziesiętnej umożliwia wyspecyfikowanie reprezentacji wszystkich lub określonych zmiennych numerycznych w konwencji dziesiętnej, stałoprzecinkowej. Do specyfikacji reprezentacji wszystkich zmiennych służy następująca instrukcja:

OPTION ARITHMETIC FIXED **liczba.liczba_cyfr_po_kropce**

określająca liczby cyfr przed znakiem kropki oraz po kropce. Podobną postać ma instrukcja deklaracji zmiennej o reprezentacji stałoprzecinkowej:

DECLARE NUMERIC **identyfikator * liczba.liczba**

Wprowadzono więc instrukcję deklaracji, podobnie jak w innych językach wysokiego poziomu — czyli to, przed czym BASIC bronił się najdłużej. Możemy jednak przyjąć, że jest to moduł dla bardziej zaawansowanych programistów, szczególnie zajmujących się oprogramowaniem algorytmów z zakresu księgowości.

REDAGOWANIE PROGRAMÓW

Moduł redagowania ma mechanizmy umożliwiające tworzenie i modyfikowanie programów. Został on zaprojektowany specjalnie z myślą o języku BASIC i jako jedyny moduł języka jest dostępny w fazie przygotowania programu.

Linie programu zapisanego w BASICU mogą być wprowadzane w dowolnej kolejności, natomiast moduł redagowania tworzy zestaw uporządkowany według następujących reguł:

1) program jest tworzony zgodnie ze wzrastającą numeracją linii

2) w przypadku wystąpienia dwóch linii o tych samych numerach, w programie pozostaje linia później wprowadzona.

Integralną częścią modułu są także cztery dyrektywy operujące tekstem programu:

- DELETE — usunięcie wskazanych linii
- EXTRACT — usunięcie wszystkich linii z wyjątkiem wymienionych w dyrektywie
- LIST — wydrukowanie wskazanych linii
- RENUMBER — zmiana numerów dla wymienionych linii.

Przykłady użycia poszczególnych dyrektyw podano poniżej:

```
DELETE 20, 30, 50

EXTRACT 100 TO 500, 600
LIST 50 TO LAST
RENUMBER   50 TO 200 STEP 5 AT 500
```

* * *

Język BASIC, przedstawiony w kolejnych trzech artykułach, jest już częściowo zatwierdzony jako norma lub też jest propozycją takiej normy. Można w nim wyróżnić kilka poziomów rozszerzeń, umożliwiających łatwe nauczanie języka oraz wykorzystywanie — zależnie od potrzeb i umiejętności użytkownika. Istotne jest to, że po wielu latach ten najpopularniejszy na Zachodzie język programowania doczeka się normy.

Warto dążyć do tego, aby także w Polsce każda implementacja tego języka była zgodna z normą. Najwyższy bowiem czas, aby ograniczając nieco pomysłowość uzyskać jednolite produkty na wielu maszynach, co umożliwi łatwe przenoszenie programów i przekazywanie gotowych prac. Dlatego też należałoby — zaraz po zatwierdzeniu standardu przez ISO — podjąć prace nad zatwierdzeniem polskiej normy języka.

ANDRZEJ FABIJANEK
KRZYSZTOF MARCINIAK
JERZY WOJCIECHOWSKI

Instytut Automatyki Przemysłowej
Politechnika Warszawska

LITERATURA

[1] Draft American National Standard for BASIC. Report ANSI X3J2/82—17, American National Standards Institute, New York, October 1982

[2] Industrial Real-Time BASIC. Report EWICS TC2 81/8. European Workshop an Industrial Computer Systems, Brussels, September 1981

[3] Real-Time BASIC for CAMAC. ESONE Committee, Luxemburg, September 1981

GOBI – system do obliczeń geometrycznych

Projekt GOBI jest próbą systematycznej organizacji opisu danych i zasad programowania urządzeń mechanicznych sterowanych numerycznie, takich jak: frezarki, tokarki, urządzenia do cięcia blach, plottery, manipulatory, roboty przemysłowe itp.

W opracowanym systemie do stałej części podstawowej dołączane są wymienne bloki procedur specjalistycznych.

CZĘŚĆ STAŁA

Część stała umożliwia tworzenie następujących konstrukcji językowych:
— definicji stałych
— definicji typów tablicowych i rekordowych
— deklaracji zmiennych
— instrukcji warunkowych, cyklu oraz skoku
— deklaracji podprogramów.

Dopuszczalne jest rekurencyjne wywoływanie podprogramów. Parametry mogą być wywołane przez wartość lub adres. LICZBA, SŁOWO i TEKST są jedynymi typami standardowymi. Z nazwą symboliczną kojarzony jest typ domyślny LICZBA.

Jako przykład mechanizmów językowych dostępnych w części stałej rozpatrzmy program:

```
PROGRAM STAL

STALE PI=3.141592
      N=5
TYPY  PUNKT=(WSPX,WSPY)
      OKRĄG=(ŚRODEK:PUNKT, PROMIEŃ:LICZBA)
ZMIENNE
      I,J:LICZBA
      A:TABLICA(N):OKRĄG
PROGRAM SILNIA(I)
      GDY I<=1; SILNIA=1 INACZEJ SILNIA=I*SIL-
      NIA(I—1)KON
KON
POCZ
      DLA I=1,1,N
            A(I)=PUNKT(I,I*PI),1/SILNIA(I)
      KON
      PISZ(A)
KON
```

W programie STAL zdefiniowano stałe PI oraz N o wartościach odpowiednio 3.141592 oraz 5. Następnie umieszczono definicję rekordu o nazwie PUNKT i dwu polach WSPX, WSPY typu LICZBA. Po niej następuje definicja rekordu OKRĄG o dwóch polach ŚRODEK typu PUNKT i PROMIEŃ typu LICZBA. Po definicjach następuje deklaracja zmiennych I, J typu LICZBA oraz A typu TABLICA (pięcioelementowa), składająca się z elementów typu OKRĄG. Ostatnia deklaracja opisuje podprogram SILNIA(I), obliczający wartość I!. Symbol POCZ rozpoczyna część operacyjną programu, która nakazuje dla I=1 z krokiem 1 do N wykonać:
— polu ŚRODEK zmiennej A(I) przypisać wartość będącą punktem o współrzędnych (I,I*PI)
— polu PROMIEŃ zmiennej A(I) typu OKRĄG przypisać wartość 1/SILNIA(I). Po wykonaniu pętli wydrukować wartość A.

PRZEBIEG PRZETWARZANIA

Przetwarzanie zadania odbywa się w dwóch etapach:

- translacja programu z postaci źródłowej do prefiksowej postaci pośredniej:

$0_n, a_1 \ldots a_n$

gdzie:
0_n — operacja n-argumentowa, a_i — i=1...n adresy pól argumentów lub operatory (wraz ze swymi argumentami) dostarczające adres argumentu.

- interpretacja programu w postaci pośredniej.

DOSTOSOWANIE PAKIETU DO POTRZEB UŻYTKOWNIKA (CZĘŚĆ ZMIENNA)

Modyfikacja pakietu polega na zmianach translatora, a w przypadku wprowadzania nowych operatorów i procedur standardowych również programu interpretującego. Modyfikacja translatora następuje poprzez wymianę jego

tablic, wykonywaną przez GENERATOR WYDANIA na podstawie danych opisujących wydanie pakietu. W ten sposób można umożliwić translatorowi akceptowanie nowych stałych, zmiennych, typów i operatorów, a także — programów standardowych.

Rozpoznanie przez translator nowych operatorów i procedur standardowych musi pociągać za sobą uzupełnienie programu interpretującego o procedury opisujące znaczenie rozpoznawanych konstrukcji. Interpreter napisano tak, że dołączeniu procedury zachowującej reguły komunikacji z otoczeniem towarzyszy praktycznie tylko dopisanie jej wywołania do programu interpretującego.

ZASTOSOWANIA

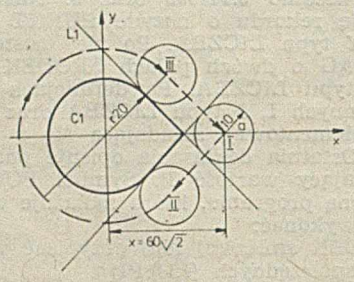
Programowanie frezarek sterowanych numerycznie

Pakiet zastosowano między innymi do progamowania frezarek sterowanych numerycznie z układami sterowania typu 2CL, wyposażając go w odpowiedni zestaw obiektów i operacji standardowych. Jako przykład prostego zadania dla frezarki może służyć program:

```
PROGRAM OBRÓBKA

      ZMIENNE L1:PROSTA,C1:OKRĄG,X
POCZ
      X=60*SQRT(2)
      C1=OKRĄG(PUNKT(0,0),20)
      L1=PROSTA(45,20)
      START(X,0,0)
      DO (C1,LEWO, STYCZNIE)
      ZA(L1,PRZÓD,C1)
      NA(PUNKT(X,0))
KON
```

Program OBRÓBKA opisuje frezowanie kształtu zaznaczonego grubą linią za pomocą freza reprezentowanego przez koło a (rys. 1). Kształt zostanie wyfrezowany poprawnie, jeśli narzędzie wykona ruch wzdłuż toru zaznaczonego linią przerywaną (przechodzącą przez położenia pośrednie I—II—III—I).



Rys. 1. Frezowanie

[I — NA(PUNKTY(X,O))
II — DO(C1,LEWO,STYCZNIE)
III — ZA(L1,PRZOD,C1)]

W programie zdefiniowano wartość zmiennych X,L1, C1 (w sposób przedstawiony na rysunku 1):

— określono położenie początkowe narzędzia (START(X,0,0))

— nakazano ruch z położenia I do II (stycznie z lewej strony do okręgu C1) (DO/C1,LEWO,STYCZNIE)

— nakazano ruch z II do III (w przód, wzdłuż okręgu C1, aż do minięcia prostej L1 całym obwodem freza) (ZA/L1,PRZÓD,C1)

— nakazano powrót do położenia I (po najkrótszej drodze, aż do osiągnięcia przez oś freza współrzędnych (X.0) (NA/PUNKT(X,0)

Wykorzystano typy standardowe PUNKT, PROSTA, OKRĄG, stałe tekstowe LEWO, STYCZNIE, PRZÓD oraz procedury standardowe DO, NA ZA, START. Wynikiem interpretacji programu jest taśma sterująca frezarką, pozwalająca na automatyczne przeprowadzenie obróbki.

Programowanie manipulatorów

W innej wersji pakiet został dostosowany do programowania manipulatorów. Wprowadzono typy standardowe:

```
PUNKT=TABLICA(3) : LICZBA
WEKTOR=TABLICA(3) : LICZBA
BAZA=TABLICA(3) : WEKTOR
UKŁAD=(PUNPO:PUNKT, BAZA:BAZA)
```

Dla typów PUNKT i WEKTOR kolejne elementy tablicy oznaczają kolejne współrzędne punktu i wektora w kartezjańskim układzie współrzędnych. Zmienna typu BAZA określa bazę przestrzeni R^3. Kolejne elementy tablicy są kolejnymi wektorami bazy. Zmienna typu UKŁAD reprezentuje pewien układ współrzędnych określony przez punkt początkowy (pole PUNPO) i bazę (pole BAZA).

Wprowadzono ponadto operatory i programy standardowe umożliwiające operowanie na zmiennych ww. typów.

Jako przykład zadania manipulacji może służyć program układania na taśmie przedmiotów wyjmowanych z palety. W programie wykorzystano m.in. programy standardowe:

UKPUN(P1,P2,P3) — wyznaczający wartość typu UKŁAD. Początek układu znajduje się w punkcie P1. Baza jest ortonormalna i prawoskrętna. Pierwsza oś jest prostopadła do płaszczyzny wyznaczonej przez punkty P1, P2, P3, a trzecia oś bazy jest zgodna z kierunkiem i zwrotem wektora P2—P1. OBWEKT(W) — wyznaczający macierz obrotu.

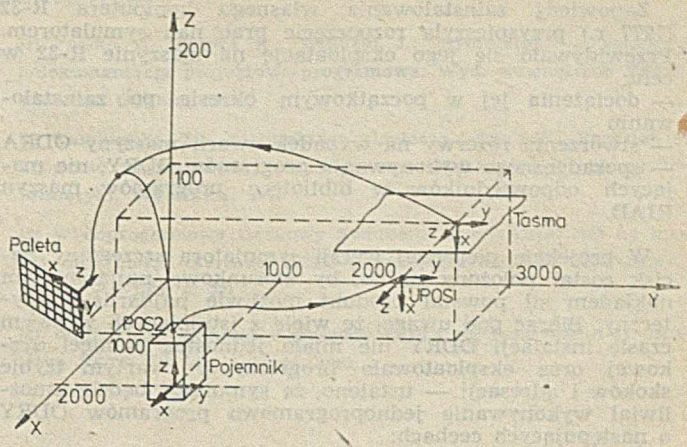
Argument W — typu WEKTOR określa kierunek obrotu, a jego długość — kąt obrotu.

Ruch łapy robota jest definiowany przy użyciu procedury DO. Ruch odbywa się z położenia aktualnego, przez położenia pośrednie, określone przez zmienne typu UKŁAD lub PUNKT, będące kolejnymi parametrami aktualnymi procedury. Stałe tekstowe WYBIEG i (lub) DOBIEG wskazują, że ruch powinien się rozpocząć i (lub) kończyć przesunięciem w kierunku wektora występującego po stałej tekstowej.

```
PROGRAM UKŁADANIE

$ UKŁADY OKREŚLAJĄCE POŁOŻENIE PALETY, TAŚ-
                                    MY, POJEMNIKA:
ZMIENNE PALETA, TAŚMA, POJEMNIK, UPOS: UKŁAD
$ POŁOŻENIE POŚREDNIE
UPOCZ: UKŁAD
$ POŁOŻENIE POCZĄTKOWE I KOŃCOWE ŁAPY RO-
                                    BOTA:
POS1,POS2:PUNKT
WX,WY,WZ:WEKTOR
PROGRAM PRZYNIES (X1,Y1)
$ PROCEDURA PRZENOSZENIA JEDNEGO PRZEDMIOTU
$ X1Y1 — WSPÓŁRZĘDNE PRZEDMIOTU W UKŁA-
                                    DZIE
$ OKREŚLAJĄCYM POŁOŻENIE PALETY
$PRZEDMIOTY PRZENOSZONE SĄ DO POJEMNIKA
                                    PRZEZ PUNKTY POS1
$ I POS2, A Z POJEMNIKA NA TASMĘ PRZEZ UKŁAD
                                    POŚREDNI UPOS.
POCZ
$ RUCH PO PRZEDMIOT Z PALETY PRZEZ POŁOŻE-
NIE POS1
DO (POS1,PALETA+X1 * WX + Y1 * WY, DOBIEG,
                                    —20 * WZ)
CHWYC
$ RUCH Z PRZEDMIOTEM DO POJEMNIKA (PRZEZ
POS1 I POS2)
DO (POS1,POS2, POJEMNIK, WYBIEG, 20 * WZ)
CZEKAJ 2
$ PRZENIESIENIE PRZEDMIOTU NA TAŚMĘ
DO (POS2,UPOS,TAŚMA)
PUSC
KON
POCZ
$ DEFINICJE UKŁADÓW I PUNKTÓW OKREŚLĄJĄ-
                                    CYCH TOR RUCHU ŁAPY
$ ROBOTA
PALETA = UKPON(PUNKT(1000,—500,300/, PUNKT(900,—
300,400), PUNKT(1100,—200,200))
```

```
$ PROCEDURA ODNIES GENERUJE OBIEKT TYPU
                              UKŁAD O WARTOŚCI
$ ODPOWIADAJĄCEJ UKŁADOWI ODNIESIENIA
POJEMNIK=ODNIES+WEKTOR (2000,1000,0)
$ UKŁAD TAŚMY JEST OKREŚLONY PRZEZ PRAWO-
                              SKRĘTNY OBRÓT O 90
$ STOPNI WOKÓŁ WERSORA (0,1,0) BAZY ODNIESIE-
NIA.
TAŚMA=PUNKT(1000,3000,1000), OBWEKT (90 * WERSY) *
BAZOD
POS1=1000,0,1500;POS2=2000,1000,500
UPOS=TAŚMA+WEKTOR(1000,0,0)
$ DEFINICJA POŁOŻENIA POCZĄTKOWEGO
UPOCZ=POS1,BAZOD
$ WX,WY,WZ WERSORY BAZY UKŁADU PALETY
WX=PALETA.BAZA(1);  WY=PALETA.BAZA(2);  WZ=
PALETA.BAZA(3)
$ OKREŚLENIE POŁOŻENIA POCZATKOWEGO LAPY
                              ROBOTA
START UPOCZ
DLA X1=0,20,200
   DLA Y1=0,30,300
      PRZENIEŚ(X1,Y1)
   KON
KON
$ POWRÓT DO POŁOŻENIA POCZĄTKOWEGO
      DO (UPOS,UPOCZ)
      STOP
      PISZ(„WYKONANIE")
KON
```



Rys. 2. Układanie na taśmie przedmiotów wyjmowanych z palety

Pakiet GOBI opracowano w standardowym podzbiorze języka FORTRAN. Uruchomiono programy na komputerach serii ODRA 1300 oraz CYBER. Dotychczas uruchomiono rozszerzenie pakietu służącego do:

- programowania frezarek SN z układami sterowania typu 2CL (1978 r., ODRA 1300)
- programowania obróbki śrub okrętowych (1980 r., ODRA 1300)
- progamowania manipulatorów (1982 r., ODRA 1300).

MAREK DZIEDZIC
KRZYSZTOF PERYCZ
JERZY WILIŃSKI

ZETO Gdańsk

Wieloprogramowy dyskowy symulator ODRY 1305 na R-32

Przed wprowadzeniem komputerów Jednolitego Systemu większość istniejących doświadczeń w projektowaniu, programowaniu i eksploatacji systemów informatycznych związana była z rodziną maszyn cyfrowych ODRA 1300. Zgodność programowa tych maszyn z komputerami ICL serii 1900 umożliwiała bezpośrednie stosowanie bogatego oprogramowania firmowego ICL oraz adaptację odnośnych technologii projektowania i programowania. Eksploatację systemów informatycznych ułatwiały systemy operacyjne GEORGE-2 i GEORGE-3.

Wieloletnie przyzwyczajenia, w początkowym okresie, stanowiły czynnik opóźniający przejście do przetwarzania na maszynach Jednolitego Systemu. Najpoważniejszym utrudnieniem był wszak brak doświadczeń praktycznych, niedostępność niektórych istotnych publikacji i odmienność kryteriów jakości przetwarzania maszyn rodziny RIAD w porównaniu do maszyn serii ODRA. W tej sytuacji było oczywiste, że przejście od maszyn ODRA do maszyn RIAD będzie zagadnieniem bardzo złożonym i wymagać będzie stworzenia szeregu narzędzi ułatwiających ten proces.

Już w 1973 roku powstał w ZETO-Gdańsk pomysł stworzenia symulatora ODRY na RIADZIE jako jednego z tych narzędzi. Został on pomyślany jako program niezależny, tj. nie wymagający do swojej eksploatacji żadnego systemu operacyjnego. Głównym celem tego programu jest wykonywanie instrukcji maszynowych ODRY na komputerze RIAD.

Program binarny ODRY — znajdujący się razem z symulatorem w pamięci operacyjnej RIADA, jest wykonywany metodą interpretacyjną. Kolejne rozkazy maszynowe ODRY są pobierane i dekodowane, a następnie — uaktywniany jest odpowiedni ciąg rozkazów maszyny RIAD, tak by otrzymać w efekcie końcowym wynik zgodny z działaniem tej samej instrukcji na maszynie ODRA.

Oczywiście, by rozpocząć wykonywanie programu ODRY, symulator musi wykonać wiele czynności organizacyjnych — takich choćby, jak załadowanie programu do pamięci operacyjnej z odpowiedniego nośnika. Aby to umożliwić, symulator wypełnia wiele funkcji systemu operacyjnego (egzekutora), jak: obsługa urządzeń wejścia-wyjścia, koordynacja procesów systemowych i programów użytkowych oraz realizacja ekstrakodów i poleceń operatorskich. Dla zapewnienia całkowitej wymienności z instalacjami ODRY — tak zbiory danych, jak i same programy binarne mają na nośnikach format i zestaw kodów maszyny ODRA.

Zapowiedź zainstalowania własnego komputera R-32 (1977 r.) przyspieszyła rozpoczęcie prac nad symulatorem. Przewidywało się jego eksploatację na maszynie R-32 w celu:
— dociążenia jej w początkowym okresie po zainstalowaniu
— stworzenia rezerwy na wypadek awarii maszyny ODRA
— sporadycznego wykonywania programów ODRY, nie mających odpowiedników w bibliotece programów maszyn RIAD.

W projekcie pierwszej wersji symulatora szczególny nacisk został położony na to, by stosunkowo najmniejszym nakładem sił powstał produkt możliwie najbardziej użyteczny. Biorąc pod uwagę, że wiele z istniejących w owym czasie instalacji ODRY nie miało jednostek pamięci dyskowej oraz eksploatowało programy w zwartym trybie skoków i adresacji — ustalono, że symulator będzie umożliwiał wykonywanie jednoprogramowo programów ODRY o następujących cechach:
— niezaufanych
— jednoczłonowych
— działających w zwartym trybie skoków i adresacji
— nie używających instrukcji BCT, BFP, SMO i MVCH
— nie współpracujących z urządzeniami zewnętrznymi w trybie odpowiedzi bezpośredniej
— nie używających jednostek pamięci dyskowej.

W praktyce najpoważniejsze ograniczenie stanowił punkt ostatni, bowiem przeważająca większość programów eksploatowanych na maszynie ODRA spełnia pozostałe wymagania.

Podstawowe prace nad tak zdefiniowanym symulatorem zostały zakończone w roku 1978 [1]. ZETO-Gdańsk sprzedało tę wersję symulatora kilkunastu użytkownikom, z których kilku eksploatowało ją jeszcze do niedawna. Uwagi użytkowników przyczyniły się do wyeliminowania pewnej liczby usterek i stanowiły zachętę do dalszego rozwoju symulatora, zwłaszcza w kierunku umożliwienia jego współpracy z jednostkami pamięci dyskowej. Okazało się bowiem, że zainteresowanie symulatorem nie zmalało, zwłaszcza wśród użytkowników dysponujących wyłącznie maszynami RIAD i eksploatujących równolegle systemy użytkowe (bądź tworzących takie) na ODRACH w ośrodkach obcych.

W początkowej fazie prac nad symulatorem autorzy często spotykali się z poglądem, że nie da się go w ogóle wykonać. Miało to wpływ na przyjęty zakres prac, który dzisiaj może wydać się skromny. Wykonanie jednak tej wersji symulatora okazało się niezbędnym krokiem umożliwiającym pomyślną realizację jego nowej wersji.

NOWY SYMULATOR

Wiosną 1982 r. zapadła decyzja realizacji nowego symulatora, nie zawierającego już żadnego z ograniczeń poprzedniej jego wersji. Głównym celem przy ustalaniu założeń projektowych było osiągnięcie całkowitej zgodności programowej z maszyną ODRA 1305 pracującą pod kontrolą egzekutora E6RM. Oznaczało to w praktyce konieczność napisania symulatora nieomal od nowa. Prace nad nim zostały zakończone późną jesienią 1982, a do końca tego roku ZETO-Gdańsk sprzedało dwa jego egzemplarze, które są już eksploatowane.

Do realizacji przyjęto wersję symulatora uwzględniającą obsługę najczęściej spotykanych w zestawach maszyn R-32 następujących urządzeń peryferyjnych:
- konsoli operatorskiej (jednej)
— czytnika kart (maksimum trzech)
— perforatora kart (jednego)
— pamięci taśmowej (maksimum ośmiu jednostek)
- pamięci dyskowej 30 MB lub 8 MB (maksimum sześciu jednostek).

Wersja ta umożliwiająca m.in. wykonywanie programów ODRY działających w zwartym lub rozszerzonym trybie adresacji i skoków, także wieloczłonowych i zaufanych typu GEORGE-2, a w szczególności programów standardowych ODRY, takich jak: translatory, programy organizacyjne, sortujące itp. Liczba i wielkość jednocześnie wykonywanych (lub znajdujących się w pamięci operacyjnej) programów jest ograniczona tylko wielkością pamięci operacyjnej maszyny RIAD. I tak, w przypadku pamięci operacyjnej o pojemności 256 KB — możliwe jest wprowadzenie do niej programów ODRY o łącznej wielkości ok. 50 K słów (ponad 110 K słów przy pamięci operacyjnej o pojemności 512 KB).

Użyta technika symulacji umożliwia natychmiastowe przenoszenie całych systemów informatycznych (biliotek programów i zbiorów danych) z maszyny ODRA na maszynę RIAD i odwrotnie. W przypadku kart perforowanych czy taśm magnetycznych przenoszenie to jest bezpośrednie, natomiast w przypadku dysków magnetycznych — odbywa się ono za pośrednictwem taśmy magnetycznej. W tym celu przy użyciu programu ≠XPJW składuje się zawartość pakietu dyskowego na taśmie magnetycznej, a następnie — przy użyciu tego samego programu — odtwarza się pakiet na innej maszynie.

Brak bezpośredniej wymienności zbiorów przechowywanych na pakietach dyskowych spowodowany jest zasadniczymi różnicami konstrukcyjnymi napędów dyskowych. Spowodowało to konieczność symulowania jednostek pamięci dyskowej ODRY na jednostkach pamięci dyskowej typu RIAD. Symulowane są wszystkie operacje używane przez egzekutor E6RM dla dysków typu EDS 8, dzięki czemu logiczna organizacja dysków w pełni odpowiada normom UDAS.

Po zaformatowaniu, pakiet 30 MB ma efektywną pojemność 22 mln znaków ODRY, a pakiet 8 MB — 5,6 mln znaków.

Obsługa symulatora jest identyczna z obsługą egzekutora E6RM na maszynie ODRA 1305. W szczególności — formaty zarówno komunikatów symulatora, jak i poleceń operatorskich są identyczne z ich odpowiednikami na ODRZE. Nie ma więc potrzeby przeszkalania operatorów ODRY pod kątem pracy z symulatorem na maszynie RIAD.

Ze zrozumiałych względów programy ODRY wykonywane są na RIADACH pod symulatorem wolniej (straty na symulację). Na czas wykonywania programów decydujący wpływ wywiera jednak nie szybkość jednostki centralnej, ale szybkość używanych przez te programy urządzeń wejścia-wyjścia, a także — w przypadku krótkich przebiegów — częstość i czasochłonność interwencji operatorskich. Jest to tak ściśle związane z charakterem i konstrukcją wykonywanych programów (systemów), że w wielu przypadkach szacowanie prędkości przetwarzania nie jest możliwe bez konkretnych prób.

Możliwe jest natomiast porównanie prędkości działania starej i nowej wersji symulatora. W starej wersji operacje wejścia-wyjścia były synchroniczne, a poza tym symulator działał jednoprogramowo. Nowa wersja stosuje asynchroniczne operacje wejścia-wyjścia i wieloprogramowość, co umożliwia nakładanie się czasów przesłań i przetwarzania. Następuje więc wyraźne (1,5—2-krotne) przyspieszenie wykonywania programów dokonujących wielu operacji wejścia-wyjścia.

Dodatkowo, obsługiwanie przez nowy symulator pamięci o bezpośrednim dostępie umożliwiło zastosowanie bardziej ekonomicznych i szybszych programów sortujących (np. ≠XSDP zamiast ≠XSMC), co dało ok. dwukrotne przyspieszenie sortowania.

Niestety, istota symulacji nie pozwala na dalsze zwiększanie szybkości działania symulatora.

DALSZY ROZWÓJ SYMULATORA

Do tej pory rozwój symulatora polegał na rozszerzaniu zakresu symulacji w stronę osiągnięcia pełnej zgodności programowej z maszyną ODRA 1305. Obecnie, gdy zostało to już spełnione, tematem prac jest skrócenie czasów wykonywania instrukcji ODRY. Ponieważ możliwości optymalizacji czasowej kodu symulatora zostały wyczerpane, jedyną drogą umożliwiającą takie przyspieszenie jest wybór jego krytycznych czasowo fragmentów, a następnie — umieszczenie w formie mikroprogramów w pamięci stałej maszyny R-32. Operacja taka efektywnie przekształca symulator w emulator, nad którym prace w ZETO-Gdańsk już trwają, z przewidywanym zakończeniem pod koniec bieżącego roku.

[1] W skład zespołu autorskiego wchodzili: mgr Zenon Fręśko mgr inż. Krzysztof Perycz i mgr inż. Jerzy Willński

Podobnie jak nowa wersja symulatora nie mogłaby powstać, gdyby wcześniej nie została zrealizowana jego wersja pierwotna, tak trudno sobie wyobrazić tworzenie efektywnego emulatora bez zebrania doświadczeń z eksploatacji nowej wersji symulatora. W pierwszej kolejności zostanie zaprogramowana w postaci mikroprogramów pętla pobierania i dekodowania rozkazów ODRY, a następnie niektóre rozkazy, których symulacja przyczynia się do powstania największych strat czasowych. Tego typu podejście umożliwia stosunkowo największe przyspieszenie wykonywania instrukcji ODRY przy względnie małej ingerencji w układy logiczne komputera R-32, a więc niewielkim kosztem.

Dokładniejsze oszacowanie stopnia przyspieszenia wykonywania programu ODRY pod nadzorem emulatora, w porównaniu do symulatora, będzie możliwe dopiero w zaawansowanym stadium prototypu.

LITERATURA

[1] Fręśko Z., Perycz K., Wiliński J.: Symulator — założenia i dokumentacja projektowo-programowa. Wyd. wewnętrzne ZETO-Gdańsk, Gdynia, 1978

[2] Kamburelis T.: Architektura logiczna emc JS. Rozdział 9: Zgodność programowa i emulacja emc ODRA 1305. Problemy Informatyki, Warszawa, 1976

[3] Wieloprogramowy Dyskowy Symulator m.c. ODRA 1300 na m.c. J.S. Informacja ofertowa ZETO-Gdańsk, Gdynia, 1982 (INFORMATYKA, nr 1/83, s. 32)

[4] Wiliński J.: Symulator maszyny ODRA na maszynie R-32. Biuletyn ZETO-Gdańsk, Gdynia, 1978.

PIOTR PACHOWICZ
RYSZARD TADEUSIEWICZ

Akademia Górniczo-Hutnicza
Kraków

CESARO – system analizy i rozpoznawania obrazów wizualnych

W poniższym artykule przedstawimy strukturę sprzętową oraz programową Cyfrowego Eksperymentalnego Systemu Analizy i Rozpoznawania Obrazów CESARO, który powstał w wyniku współpracy Instytutu Automatyki, Inżynierii Systemów i Telekomunikacji [1] oraz Instytutu Przeróbki i Wykorzystania Surowców Mineralnych AGH. O eksperymentalnym charakterze systemu decyduje jego wysoki koszt oraz wprowadzenie próbnych rozwiązań niektórych zagadnień.

Liczba systemów analizy obrazów wizualnych w Europie [3] nie przekracza dwudziestu instalacji, z czego w Polsce dostępne są dwie. Możliwości obliczeniowe, jakie dają tego typu systemy, pozwalają na automatyzację operacji, przy których człowiek wykorzystuje wzrok, a także na szybszą ocenę charakterystycznych wielkości analizowanych obiektów oraz rozpatrywanie współczynników praktycznie niemożliwych do obliczenia przez człowieka.

System CESARO przeznaczony jest do analizy obrazów w zakładach przemysłowych, a zwłaszcza w przemyśle wydobywczym. Jego własności pozwalają na dowolne inne zastosowania, choćby w przemyśle lekkim (np. włókienniczym), w medycynie czy biologii.

STRUKTURA SPRZĘTOWA

System składa się z zespołu urządzeń przetwarzających sygnał wizualny w sygnał cyfrowy oraz z komputera analizującego ten sygnał (rys. 1). Zespół wprowadzania obrazów do komputera wyposażony jest w kamerę telewizji przemysłowej, której, sygnał przetwarzany jest w szybkim przetworniku analogowo-cyfrowym, a następnie — w postaci sygnału cyfrowego — gromadzony w pamięci buforowej.

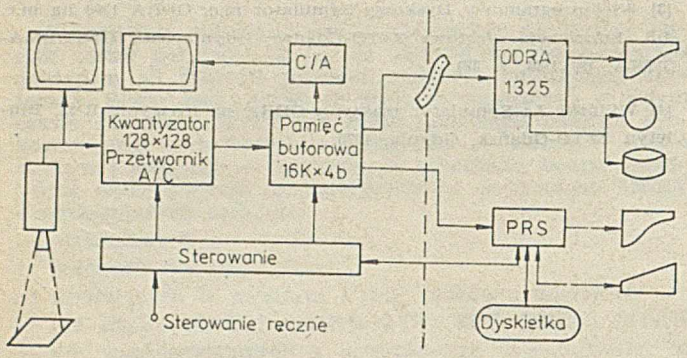
Wydzielona część pola widzenia kamery zostaje zamieniona w matrycę o wymiarach 128×128 punktów tworzą-

[1] Adres do korespondencji: Instytut Automatyki, Inżynierii Systemów i Telekomunikacji, Akademia Górniczo-Hutnicza, Al. Mickiewicza 30, 30-059 Kraków

Mgr inż. PIOTR PACHOWICZ ukończył w 1981 roku Wydział Elektrotechniki, Automatyki i Elektroniki AGH w Krakowie i podjął pracę w Zakładzie Biocybernetyki Instytutu Automatyki, Inżynierii Systemów i Telekomunikacji AGH. Zajmuje się problemami modelowania i symulacji obiektów biologicznych oraz metodami i technikami automatycznego rozpoznawania obrazów.

Doc. dr hab. inż. RYSZARD TADEUSIEWICZ jest absolwentem Wydziału Elektrotechniki, Automatyki i Elektroniki AGH w Krakowie. Zasadniczą domeną jego zainteresowań są metody i techniki automatycznego rozpoznawania obrazów i mowy. Z tego zakresu obronił pracę magisterską (1971), doktorską (1975) i habilitacyjną (1980). Od 1973 roku kieruje Samodzielną Pracownią Biocybernetyki, a od 1980 jest zastępcą dyrektora Instytutu Automatyki, Inżynierii Systemów i Telekomunikacji AGH.

cych kwadratowy raster. Natomiast pojedynczemu punktowi w wyniku dyskretyzacji przypisuje się liczbę z zakresu 0—15 określającą stopień intensywności oświetlenia. Dla prawidłowej kontroli położenia i odpowiedniego oświetlenia obiektów w kadrze wykorzystywane są dwa monitory telewizyjne, wyświetlające obraz bezpośrednio z kamery oraz po przetworzeniu w ciąg impulsów cyfrowych.



Rys. 1. Schemat struktury sprzętowej systemu CESARO

W wersji eksperymentalnej systemu dane z pamięci buforowej zostają wyprowadzone przez perforator taśmy papierowej i przeniesione na komputer ODRA 1325. Natomiast przy zastosowaniu specjalistycznym — pamięć buforowa ma bezpośrednie połączenie z produkowanym seryjnie minikomputerem górniczym PRS. (p. INFORMATYKA nr 8—9/82, s. 19 — przyp. red.). Sprzężenie zespołu wprowadzania obrazów z minikomputerem pozwala na osiągnięcie integralności systemu oraz na pracę w trybie bezpośrednim (ang. on-line). Zastosowanie małego minikomputera, jakim jest PRS — Przemysłowy Rejestrator Skażeń, zmniejsza poza tym, koszty systemu.

Mając na względzie nowatorstwo metod stosowanych w przetwarzaniu obrazów wizualnych, jak i doskonalenie technik przetwarzania, przewidziana została także współpraca w trybie rozłącznym (ang. off-line) z ODRĄ 1300. Dzięki temu jedynie dopracowane metody poszczególnych operacji na obrazach, jak i obliczeń ukierunkowanych na wybrane zagadnienia, zostają ujęte w specjalistyczne pakiety programowe pracujące w trybie konwersacyjnym na minikomputerze.

PROGRAMOWE PRZETWARZANIE OBRAZÓW

Ogólny schemat programowego przetwarzania obrazów przedstawiony został na rysunku 2. Na pięciu poziomach wyróżnione są grupy procedur (napisanych w języku PLAN i FORTRAN, realizujących specyficzne zadania.

Podstawową grupę procedur określonych na pierwszym poziomie stanowią podprogramy kodowania i dekodowania obrazów. Konieczność używania tych procedur wynika z upakowania w jednej komórce pamięci o długości 24 bitów po 6 punktów rastra obrazu tonowanego (każdy punkt zajmuje 4 bity, co pozwala na odróżnienie 16 poziomów szarości). W przypadku obrazów binarnych, tj. takich, w których pojedynczemu punktowi rastra przypisuje się wartości 0 lub 1, w jednej komórce pamięci operacyjnej upakowane są 24 punkty. Taki sposób upakowania znacznie zwiększa czas obliczeń, lecz umożliwia równoczesne umieszczenie w pamięci operacyjnej trzech obrazów tonowanych i trzech obrazów binarnych.

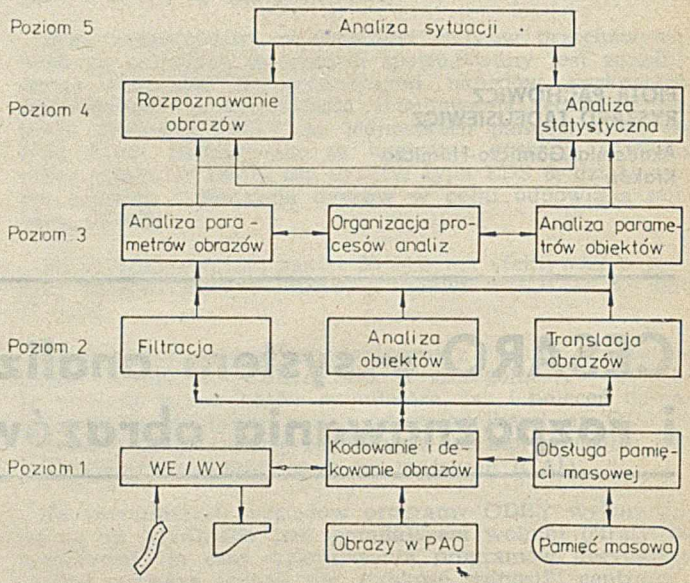
Oddzielną grupę procedur pierwszego poziomu tworzą podprogramy wprowadzania obrazów z taśmy dziurkowanej oraz ich wydruku w rozmaitych formach na drukarce wierszowej.

Ostatnią podstawową grupę procedur tworzą podprogramy manipulacji obrazami w pamięciach masowych. Należą do nich podprogramy archiwizacji, aktualizacji, wyszukiwania wyszczególnionych obrazów i wyprowadzania ich do pamięci operacyjnej komputera.

Na drugim poziomie wyszczególniona została pokaźna grupa procedur filtracji i ogólnie pojętej eliminacji zakłóceń, występujących na drodze kamera-komputer. Stosowane są w tym celu podprogramy filtracji logicznej i uśredniającej, dyskretne dwuwymiarowe transformacje Fouriera, Wlasha, Hadamarda, określanie gradientów jasności i wydobywania konturów obrazu.

Drugą grupę procedur z tego poziomu stanowią podprogramy transformacji geometrycznej obrazów. I tak, występują tu podprogramy przesunięć, obrotu, odbić, symetrii, usuwania fragmentów obrazu itp.

Trzecią grupę procedur tworzą podprogramy poszukiwania i wydzielania poszczególnych części obrazu, określania ich położenia, maksymalnych wymiarów, średniego zaciemnienia itp.



Rys. 2. Schemat struktury programowego przetwarzania obrazów systemów CESARO

Ogólnie można stwierdzić, że procedury drugiego poziomu stanowią najliczniejszą grupę procedur systemu CESARO i każdy obraz przeznaczony do dalszych obliczeń musi zostać odpowiednio przetworzony przez dobrane podprogramy. Wraz z procedurami kodowania i dekodowania obrazów wykonują one obliczenia punkt po punkcie na całym obrazie i z tego też powodu czas ich pracy jest stosunkowo długi. Stanowią one wąskie gardło wszystkich systemów przetwarzania obrazowego.

Obrazy przetworzone przez procedury drugiego pozomu analizowane są za pomocą procedur poziomu trzeciego. Dla zastosowań w automatyzacji procesów przeróbki i wykorzystania surowców mineralnych utworzono na tym poziomie bogatą bibliotekę podprogramów, umożliwiających obliczanie niekiedy bardzo złożonych — z matematycznego punktu widzenia — miar wielkości ziaren, takich jak: średnice Martina, Fereta, cięciwowe, sitowe, estymowane z powierzchni, obwodu i objętości ziaren, a także rozmaicie definiowanych współczynników kształtu.

Na bazie podprogramów obliczających charakterystyczne współczynniki dla rozpoznanych obiektów analizowanego obrazu, zbudowane zostały trzy specjalistyczne pakiety podprogramów: MINERAL, ANALIZA i KSZTAŁT:

- MINERAL stosowany jest do obliczania parametrów i współczynnkiów charakterystycznych dla obrazów trójwymiarowych, zarejestrowanych i przetworzonych w trzech kolejnych rzutach; na przetwarzanym obrazie może wystąpić co najwyżej dwadzieścia obiektów
- ANALIZA dokonuje analizy obiektów dwuwymiarowych preparatów mikro- i makroskopowych, przy ograniczeniu liczby obiektów do 32 na jeden obraz
- KSZTAŁT umożliwia obliczenie najbardziej żmudnych i pracochłonnych współczynników kształtu dla obiektów dwuwymiarowych.

Każdy z wymienionych pakietów stanowi integralną całość i jest przeniesiony na system minikomputerowy współpracujący z kamerą telewizyjną w trybie bezpośrednim. Tworzenie zaś nowych pakietów dla innych zastosowań wymaga jedynie dołączenia do szkieletu programowego nowych procedur obliczeniowych, określających charakterystyczne wielkości w oparciu o analizę poszczególnych obiektów obrazu.

Obliczone przez procedury trzeciego poziomu wielkości charakteryzujące poszczególne obiekty można, dzięki procedurom poziomu czwartego, poddać obróbce statystycznej lub procesowi rozpoznawania. Procedury tego poziomu stanowią odrębną, bogatą bibliotekę podprogramów systemu CESARO.

ZASTOSOWANIA

Jak uprzednio wspomnieliśmy, system CESARO w obecnej wersji skonstruowany został z myślą o zastosowaniu w automatyzacji procesów przeróbki i wykorzystania surowców mineralnych, a w szczególności procesów wzbogacania rud [1, 2, 4]. Zastosowanie kamery telewizyjnej oraz opracowanego systemu pakietów obliczeniowych umożliwia analizę składu ziarnowego preparatów mikro- i makroskopowych w różnych formach. Dzięki wykorzystaniu szybkich metod komputerowych możliwe stało się też obliczenie skomplikowanych współczynników (dotychczas pomijanych ze względu na trudność obliczeń), które pozwalają na istotne poszerzenie badań naukowych w omawianym zakresie.

Przeprowadzone zostały również wstępne badania nad zastosowaniem systemu CESARO w sterowaniu procesem flotacji (wzbogacania), głównie rud metali, na podstawie optycznej oceny piany flotacyjnej. Uzyskane wyniki wykazują, że dzięki systemowi sterowanie procesem może być trafniejsze i tańsze, niż w przypadku dokonywania oceny jakichkolwiek własności fizycznych lub chemicznych cieczy flotacyjnej.

*

Badania z zakresu komputerowej analizy i rozpoznawania obrazów wizualnych, prowadzone w wielu laboratoriach świata, dotyczą różnych dziedzin. Do głównych zastosowań tych systemów zaliczyć należy nauki biomedyczne, sterowanie ruchem ulicznym, rozpoznawanie fotografii twarzy, odcisków palców, sterowanie robotami przemysłowymi oraz analizowanie zdjęć satelitarnych i lotniczych [3]. Szczególne perspektywy zastosowań gospodarczych pojawiają się w sterowaniu procesami przemysłowymi, gdzie występuje szczególna uciążliwość pracy lub niebezpieczeństwo utraty zdrowia ludzkiego. Wykorzystywanie w procesie sterowania informacji obrazowej stwarza zupełnie nowe możliwości dla szeroko rozumianej inżynierii systemów.

PERSPEKTYWY SYSTEMU

Jak uprzednio zaznaczyliśmy, proces przetwarzania obrazowego na niektórych etapach jest szczególnie czasochłonny, co wynika z szeregowego (punkt po punkcie) przetwarzania rastra obrazu oraz konieczności każdorazowego rozpakowywania i upakowywania punktów w komórkach pamięci komputera. Niemniej konieczność upakowania punktów rastra istotnie zmniejsza koszty stosowania dużych pamięci komputera.

Dla ominięcia tych wad, laboratoria na świecie pracują nad nową generacją komputerów opartych o równoległe przetwarzanie kolejnych wycinków obrazu przez wiele procesorów. Stosowanie jednak tego typu rozwiązań znacznie zwiększa koszt systemów, co czyni je mało użytecznymi dla zastosowań gospodarczych. Czasy obliczeń poszczególnych procedur filtracji, transformacji oraz procedur wyższych poziomów (rys. 2) przetwarzania obrazowego szczególnie zależą od wymiarów utworzonego rastra.

Wydaje się, że dla celów sterowania, jak i obliczeń podstawowych parametrów wydzielonych obiektów, zastosowanie rastra o wymiarach 128×128 punktów (każdy w szesnastostopniowej skali szarości) jest w pełni wystarczające i stanowi kompromis między rozmiarem zajmowanej pamięci, czasem obliczeń a ich dokładnością. Toteż odrzucony został projekt zmniejszenia wymiarów rastra dla uzyskania większych możliwości obliczeniowych systemu, a uwaga autorów została skupiona na możliwości programowego przetwarzania półrównoległego. Przetwarzanie to w obecnej chwili stanowi podstawowy kierunek doskonalenia systemu, a — ogólnie rzecz biorąc — polega na wykorzystywaniu pewnej grupy operacji na całych komórkach zawierających upakowane punkty rastra, a nie na pojedynczych punktach, jak wykonywane jest to tradycyjnie. Zastosowanie techniki półrównoległego przetwarzania, w pierwszych próbach konstrukcji szczególnie czasochłonnych procedur filtracji, ponad 30-krotnie zmniejszyło czas obliczeń.

Istotnym kierunkiem doskonalenia systemu CESARO jest adaptacja istniejących pakietów analizy obrazów na system minikomputerowy, działający w trybie bezpośrednim z kamerą telewizyjną oraz tworzenie nowych pakietów analiz.

Dodatkowo konstruowane są nowe procedury zaliczane do grup niższych poziomów. Zapotrzebowanie to wiąże się ze specyficznym wyglądem preparatów mikroskopowych, w których obiekty mają nierównomierne oświetlenie (optyczne „dziury") lub nachodzą na siebie. Zagadnienia te stanowią jednak nadal jeden z trudniejszych problemów omawianych w literaturze poświęconej rozpoznawaniu i przetwarzaniu obrazów wizualnych.

* * *

Autorzy uważają, że warto podjąć prace nad ustaleniem potencjalnych możliwości zastosowań systemu CESARO w innych dziedzinach nauki oraz procesach produkcyjnych. Liczą na kontakt z zainteresowanymi Czytelnikami, służąc poradą oraz swym czasem.

LITERATURA

[1] Kordek J., Nipl R., Sztaba K., Tadeusiewicz R.: CESARO — The Digital Experimental System of Analysis and Recognition of Images. 17th International Symposium on the Application on Computers and Mathematics in the Mineral Industries, Moscow, 1980 (pp. 393—399)

[2] Kordek J., Nipl R., Tadeusiewicz R.: Cyfrowy Eksperymentalny System Analizy i Rozpoznawania Obrazów CESARO i jego zastosowania w badaniach nad obrazami ziaren. XIII Krakowska Konferencja Naukowo-Techniczna Przeróbki Kopalin, Kraków, 1979 (s. 347—355)

[3] Kulpa Z.: Universal digital image processing systems in Europe — a comparative survey. Digital Image Processing Systems, Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, New York, 1981

[4] Tadeusiewicz R.: Programy analizy składu ziarnowego w systemie CESARO. XIII Krakowska Konferencja Naukowo-Techniczna Przeróbki Kopalin, Kraków, 1979 (s. 367—375).

Stały kontakt z INFORMATYKĄ gwarantuje tylko prenumerata

Zasady jej zamawiania — na czwartej stronie okładki

Blankiet zamówienia prenumeraty dla instytucji — na str. 53

EMILIA FLADROWSKA
Zakłady Teleelektroniczne TELKOM-TELFA
Bydgoszcz

Warunki i metody efektywnego zastosowania systemu TPP

Głównym problemem informatyki w naszym kraju jest, jak wykazano w [3], bariera praktycznych zastosowań — nieskuteczność lub mała skuteczność systemów informatycznych w praktycznym zarządzaniu przedsiębiorstwem. Jedną z głównych przyczyn tego stanu jest nieaktualność informacji zawartych w bazie danych (w zbiorach) systemu informatycznego, a zwłaszcza nieadekwatność informacji zawartych w zbiorach podsystemu Techniczne Przygotowanie Produkcji (TPP). Uniemożliwia to praktyczne wykorzystanie zbiorów TPP do określania cen, planowania, kosztów, przewidywania wyników finansowego przedsiębiorstwa itp., a więc — spełnianie praktycznych wymogów reformy gospodarczej.

Posiadanie aktualnych komputerowych zbiorów TPP jest dla przedsiębiorstw dobrodziejstwem, którego nie da się przecenić. Potwierdzają to także doświadczenia Zakładów TELKOM-TELFA.

Celem niniejszego artykułu jest pokazanie przeciętnego stopnia aktualności zbiorów technicznego przygotowania produkcji [2] i na tym tle zaproponowanie metody organizacji sprzężeń zwrotnych, wymuszających aktualność informacji w zbiorach (bazie danych) podsystemu TPP, która jest szczególnie istotna w przedsiębiorstwach o skomplikowanej strukturze wyrobów.

W artykule będzie dominował punkt widzenia Zakładów Teleelektronicznych TELKOM-TELFA w Bydgoszczy, które eksploatują podsystem TPP od 1970 roku (początkowo na komputerze MIŃSK-22, potem — na ODRZE 1305).

AKTUALNOŚĆ INFORMACJI W BAZIE DANYCH

Aktualność informacji wprowadzonych do bazy danych jest warunkiem koniecznym skutecznego stosowania systemu informatycznego w zarządzaniu przedsiębiorstwem.

Z doświadczeń zakładów TELKOM-TELFA, zbieranych już ponad dziesięć lat, wynikają następujące wnioski.

- Najłatwiej jest zapewnić aktualność baz danych systemów księgowych. Sprzyjają temu obowiązujące zasady księgowości: zasada uzgadniania kont księgowych, comiesięczny obowiązek uzgadniania ewidencji ilościowej z ewidencją ilościowo-wartościową, inwentaryzacja składników majątkowych itp.

- Każdy podsystem informatyczny, pomimo zastosowania standardowych metod kontroli, wymaga dodatkowo „własnych", specyficznych zabezpieczeń, gwarantujących aktualność informacji w bazie danych. (Przykładowo — bardzo szybko może się dezaktualizować zbiór podsystemu „Kadry". Wypracowanie układu bodźców zapewniających zgłaszanie przez pracowników przedsiębiorstwa wszystkich zmian w swoich kwestionariuszach osobowych jest problemem niezwykle trudnym).

- Utrzymanie aktualności informacji w bazie danych wymaga, obok typowych metod kontroli, takiego doprowadzenia wyników (tabulogramów) do użytkowników, które uruchomi sprzężenie zwrotne, wymuszające aktualność informacji w bazie danych systemu, czyli — „kanały zwrotnego sygnalizowania błędów".

- „Kanały zwrotnego sygnalizowania błędów" uruchomią się w takim stopniu, w jakim stopniu użytkownicy (komórki organizacyjne przedsiębiorstwa) wykorzystują wyniki systemu i w jakim stopniu wdrożenie informatycznego systemu spowoduje wyeliminowanie tradycyjnych zestawień, kartotek i innego rodzaju dokumentacji pośredniej.

- Skuteczność „kanałów zwrotnego sygnalizowania błędów" zależy od tego, w jakim stopniu użytkownicy tabulogramów wiedzą gdzie (do jakich komórek obsługujących wejście systemu) należy zgłaszać poszczególne rodzaje błędów. Istotną funkcję szkoleniową powinna w tym zakresie spełniać komórka epd.

- Największy wpływ na aktualność informacji w bazie danych mają zbiory podsystemu TPP. Należy podkreślić, że zbiory te warunkują ok. 80% obsługi systemu zarządzania przedsiębiorstwem przemysłu maszynowego.

Ze zbiorami podsystemu TPP wiążą się trzy zasadnicze problemy aktualizacji informacji:
— problem zgodności informacji zbiorów komputerowych TPP z informacjami zawartymi w źródłowej dokumentacji technologicznej (kalki przechowywane w archiwum dokumentacji technicznej)
— problem zgodności informacji wyprowadzanych w formie zestawień komputerowych z analogicznymi zestawieniami tradycyjnymi; dla przykładu: uzyskanie pełnej zgodności komputerowych i tradycyjnych zbiorczych norm zużycia materiałów na wyrób [1] jest przedsięwzięciem pracochłonnym i długotrwałym [2]
— problem zgodności technologii „pisanej" z technologią stosowaną (realizowaną w gniazdach produkcyjnych).

W procesie wdrażania podsystemu TPP i podsystemów pochodnych („planowanie produkcji", „koszty" itp.) aktualizowanie („oczyszczenie") zbioru następuje według podanej wyżej kolejności uzgadniania.

AKTUALNOŚĆ INFORMACJI W ZBIORACH PODSYSTEMU

Tradycyjna dokumentacja technologiczna zawiera różnego rodzaju błędy. Najczęściej w przedsiębiorstwach spotykane są:
— brak kompletu kalek procesów technologicznych w archiwum zakładowym oraz w komórkach, które tę dokumentację wypożyczają dla aktualizacji lub wykonania odbitek [3]

---

[1]) Zestawienia te w tradycyjnym systemie są obowiązującym składnikiem dokumentacji technologicznej i służą między innymi do planowania wielkości zużycia materiałów w dziale zaopatrzenia oraz kosztów zużycia materiałów bezpośrednich w dziale kosztów

[2]) W TELFIE proces ten trwał ok. dwa lata. W okresie tym przeniesiono indeks materiałowy z Działu Zaopatrzenia do Działu Ogólnotechnicznego, dzięki czemu uzyskano zgodność nazewnictwa materiałów z normami

[3]) Przy zakładaniu zbiorów TPP w TELFIE okazało się, że w archiwum brakuje ok. 15% kalek

— nierealne normy, a nawet brak norm zużycia materiałów i pracochłonności
— brak symboli indeksu materiałowego przy nazwach i normach zużycia materiałów
— brak symboli stanowisk i gniazd produkcyjnych wykonujących określoną detalooperację
— brak symboli stosowanych narzędzi specjalnych i uniwersalnych
— błędne symbole operacji, a nawet ich brak
— błędne rodzaje oraz ilości komponentów wchodzących w skład wyrobu.

Względnie poprawny stan dokumentacji technologicznej występuje w tych zakładach, które do wystawienia dokumentacji produkcyjnej stosują system zleceniowy i technikę powielaczy rządkujących. Natomiast w zakładach, w których pomiędzy dokumentacją technologiczną a dokumentacją produkcyjną występuje „człowiek z długopisem", źródłowa dokumentacja technologiczna dezaktualizuje się wprost proporcjonalnie do długości życia wyrobu.

Dokładniejsze niż źródłowa dokumentacja technologiczna, a więc bardziej zbliżone do rzeczywistości produkcyjnej, stają się często kartoteki lub zestawienia. Wynika to z faktu, że zauważone (zgłaszane przez produkcję) błędy koryguje się we wspomnianej dokumentacji pośredniej — odpowiednio przekształconej i będącej podstawą wystawiania dokumentacji produkcyjnej, a także planowania elementów, podzespołów i zespołów oraz zużycia materiałów. W ten sposób praktyka wymusza nieformalne uproszczenia procesu aktualizacji. Podstawową przyczyną omówionych mankamentów dokumentacji technicznej są jej rozmiary, niemożliwe do opanowania tradycyjnymi metodami. Względnie poprawny stan dokumentacji można uzyskać jedynie przy komputerowej eksploatacji podsystemu TPP oraz podsystemów pochodnych („planowanie produkcji", „koszty", „emisja dokumentacji produkcyjnej").

Zakładanie bazy danych TPP wymusza uporządkowanie źródłowej dokumentacji technologicznej, jednakże jest to raczej usunięcie formalnych niedomagań. Podstawowe problemy aktualności tej dokumentacji, tzn. zgodność zestawień komputerowych z tradycyjnymi oraz technologii „pisanej" ze stosowaną są możliwe do rozwiązania tylko poprzez eksploatację i konsekwentne stosowanie systemu informatycznego oraz skuteczną organizację wspomnianych sprzężeń zwrotnych, uruchamiających „kanały zwrotnego sygnalizowania błędów" w bazie danych.

PROPOZYCJA SPRZĘŻEŃ ZWROTNYCH W PODSYSTEMIE TPP

Sprzężenia zwrotne powodujące aktualność informacji w bazie danych podsystemu TPP powinny być zorganizowane w dwóch obszarach struktury organizacyjnej przedsiębiorstwa i realizowane w dwóch kolejnych etapach:
— w etapie zakładania i aktualizacji zbioru TPP — w obszarze działu technologicznego oraz działu informatyki;
— w etapie eksploatacji podsystemu TPP i podsystemów pochodnych („planowanie produkcji", „koszty" itp.) — w obszarze całego przedsiębiorstwa; w etapie tym powinny być zorganizowane kanały zwrotnego sygnalizowania różnic pomiędzy technologią „pisaną" i stosowaną.

Etap zakładania i aktualizacji zbioru

Rysunek 1 przedstawia podejście wypracowane ewolucyjnie i stosowane od roku 1977 w Zakładach TELKOM-TELFA. Na rysunku tym pominięto stosowane również tabulogramy kontrolne.

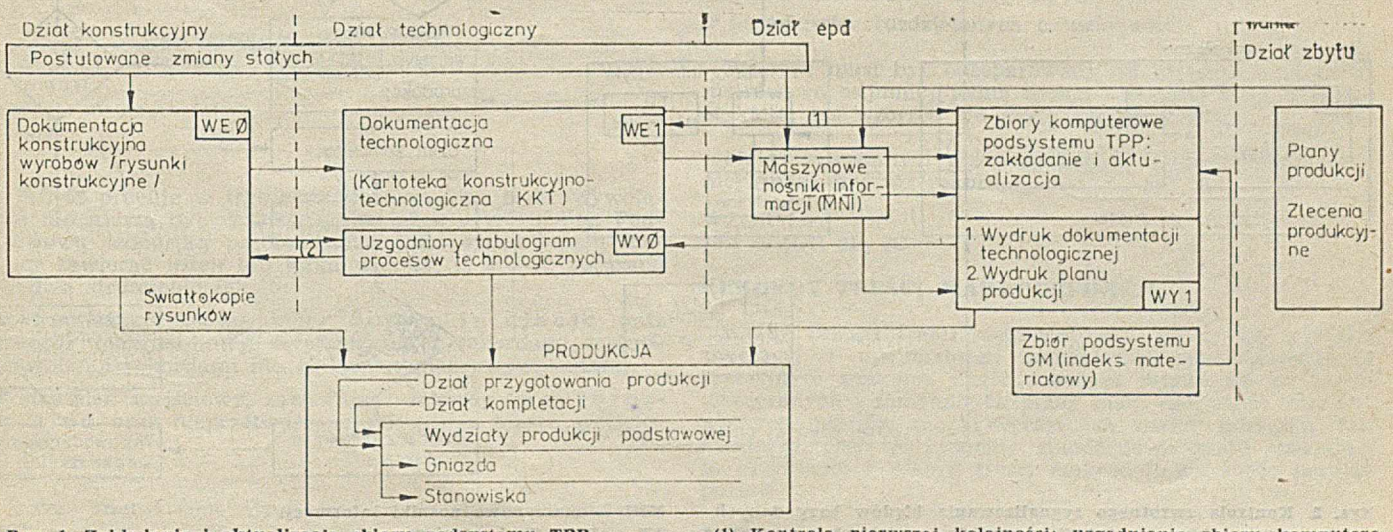
Dokumentacja konstrukcyjna produkowanych wyrobów nie zawiera wielu informacji niezbędnych w podsystemie TPP, nie może być więc ich bezpośrednim źródłem. Tradycyjna dokumentacja technologiczna, jak już wykazano, jest natomiast niekompletna, a jej układ w większości przedsiębiorstw nie gwarantuje sprawnego tworzenia maszynowych nośników informacji. Z tego względu celowe jest zaprojektowanie formularzy przystosowanych do komputerowego podsystemu TPP, a następnie przeniesienia na nie kompletu dokumentacji technologicznej produkowanych wyrobów. W przypadku nowych uruchomień, opracowanie dokumentacji powinno odbywać się już na formularzach nowo zaprojektowanych.

Takie podejście zapewnia wyeliminowanie tradycyjnej dokumentacji technologicznej, co może być zrealizowane pod warunkiem wprowadzenia tabulogramu „wydruk procesów technologicznych" (rys. 1, -WYØ)

Jak dowiodła praktyka, najwięcej trudności stwarzają przy tym podejściu:
— opracowanie katalogu operacji technologicznych, dostosowanego do specyfiki produkcyjnej przedsiębiorstwa
— zaprojektowanie wariantów tabulogramu „wydruk procesów technologicznych", dostosowanych do specyfiki produkcji poszczególnych wydziałów, a nawet gniazd produkcyjnych
— wdrożenie nowego sposobu korzystania z dokumentacji oraz jej przechowywania w archiwum centralnym oraz na wydziałach (nowa dokumentacja wprowadza zmieniony sposób zapisu, różnicę formatu i grubości papieru dokumentów).

Informacje zawarte w tabulogramie „wydruk procesów technologicznych", stanowią — przykładowo — podstawę ustawienia parametrów urządzenia na stanowisku produkcyjnym. W przypadku operacji wielozabiegowych lub operacji, przy których konieczne jest podanie wielu parametrów technicznych procesu, istnieje konieczność stosowania dla tych operacji systemu kart instrukcyjnych.

Z analizy rozwiązania prezentowanego na rysunku 1 wynikają następujące korzyści.
● Każdy wniosek zmian stałych musi być wprowadzony do zbioru TPP, ponieważ tylko tym kanałem może on przejść na produkcję.



Rys. 1. Zakładanie i aktualizacja zbioru podsystemu TPP
WE — wejścia podsystemu: Ø — tradycyjne, 1 — komputerowe
WY — wyjścia podsystemu: ϕ — kontrolno-użytkowe, 1 — użytkowe
(1) Kontrola pierwszej kolejności: uzgadnianie zbioru komputerowego z KKT (obsługuje dział epd)
(2) Kontrola drugiej kolejności: sprzężenie WE z WY podsystemu (obsługuje dział technologiczny)

● Wyeliminowanie tradycyjnej dokumentacji technologicznej zmniejsza pracochłonność obsługi wejścia podsystemu TPP, przyczyniając się tym samym do jego aprobaty. Ponadto nowa dokumentacja zmniejsza potencjalne źródła błędów (w dotychczasowym układzie służba technologiczna musi obsługiwać dwa zbiory: tradycyjny i komputerowy).

● Uzyskuje się możliwość łatwego sprawdzenia aktualności zbioru podsystemu TPP poprzez bezpośrednie porównania pierwszego wejścia systemu (WE-0,1) z pierwszym wyjściem systemu (WY-0). Sprawdzenie jedynie tych dwóch dokumentów daje pewność aktualności zbioru, a eliminowana jest przy tym konieczność sprawdzania pozostałych przekazywanych na produkcję użytkowych wydruków komputerowych. Błędy w perforacji lub nieprawidłowe wprowadzenie zmian stałych są tym sposobem natychmiast wykrywane. Przedstawiony sposób sprawdzenia treści wejścia z wyjściem jest dlatego bardzo prosty, że nie wymaga wykonywania dodatkowych przeliczeń. Jego efektywność polega na tym, że zbiory WE-0,1 i WY-0 mają zbliżoną postać oraz identyczną kolejność zapisu i poziom agregacji danych [1].

Etap eksploatacji podsystemu TPP i podsystemów pochodnych

Jak już udowodniono, w etapie zakładania i aktualizacji zbioru podsystemu istnieje możliwość doprowadzenia do pełnej zgodności informacji w komputerze oraz w dokumentacji konstrukcyjno-technologicznej. Nie oznacza to jednak, że informacje zapisane w dokumentacji oraz w zbiorze TPP w pełni odzwierciedlają rzeczywisty stan produkcji.

Praktyka wskazuje na występowanie znacznych różnic pomiędzy technologią „pisaną" i stosowaną. Wieloletnia eksploatacja systemu informatycznego w Zakładach TELKOM-TELFA udowodniła, że technologia „pisana" w takim stopniu zbliży się do technologii stosowanej, w jakim dokumentacja technologiczna w formie bezpośredniej i przetworzonej spłynie na produkcję odpowiednio zorganizowanymi kanałami komputerowymi. Z tego względu, przy przewidywaniu wdrażania wspomnianych podsystemów pochodnych należy zwracać uwagę, aby wśród strumieni informacji działały „kanały zwrotnego sygnalizowania błędów". Polega to m.in. na takim doprowadzeniu informacji do użytkowników systemu, by komórki przedsiębiorstwa korzystające z jego wyjść miały możliwość oddziaływania na komórki obsługujące wejście. Proces ten przedstawiono na rysunku 2 w odniesieniu do sfery produkcji. Analogiczny schemat postępowania występuje w sferze działów pomocniczych (zaopatrzenia, kooperacji, planowania, kosztów i cen itp.).

W Zakładach TELKOM-TELFA względną zgodność technologii „pisanej" z technologią stosowaną na produkcji uzyskano dopiero w 1976 roku, po wdrożeniu komputerowej emisji dokumentacji produkcyjnej, obejmującej:
— karty limitu (dokumenty pobrań materiałów)
— karty kompletacyjne
— wydziałowe plany produkcji części, podzespołów i zespołów
— plany potrzeb kooperacyjnych
— „wydruk procesów technologicznych" (rys. 1 WY-Ø).

Należy wyjaśnić, że poprzednio wdrożone tabulogramy z zakresu planowania, z uwagi na wysoki stopień agregacji danych oraz nikły bezpośredni kontakt z wydziałami produkcyjnymi, nie stwarzały możliwości pełnego „wyczyszczenia" dokumentacji technologicznej.

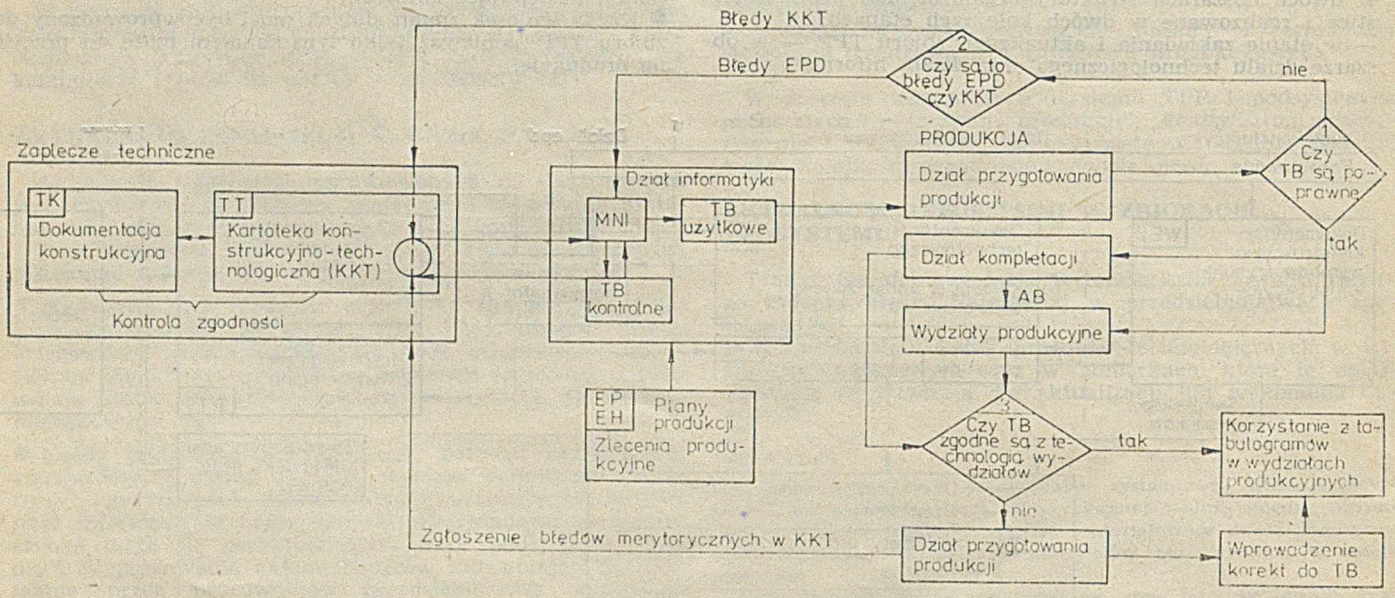
W komputerowych zbiorach kartoteki konstrukcyjno-technologicznej (KKT) w dalszym ciągu nie w pełni precyzyjna jest informacja dotycząca pracochłonności detalooperacji, z uwagi na zachowanie tradycyjnego sposobu emisji kart pracy. Należy zaznaczyć, że zbiory KKT w TELFIE zawierają ok. 64 tys. detalooperacji technologicznych i tyleż czasów jednostkowych wykonywania operacji (tj), co w pewnym stopniu usprawiedliwia istniejący stan.

* * *

Praktyka wykazuje, że w przypadku produkcji o skomplikowanej strukturze wyrobów nieosiągalne jest bez komputera precyzyjne planowanie kosztów oraz skuteczne kształtowanie cen wyrobów i wyniku finansowego przedsiębiorstwa. Jednakże rzeczywistą skuteczność wykorzystania komputera można osiągnąć jedynie przy posiadaniu aktualnych (adekwatnych do rzeczywistości) zbiorów podsystemu TPP, a także bieżącej aktualizacji cen materiałów, z których ten podsystem korzysta.

LITERATURA

[1] Dzianott A.: Organizacja wdrażania wielodziedzinowego systemu informatycznego SIKOP-1300 ze szczególnym uwzględnieniem TPP i planowania produkcji. TNOIK-PKAPI, Koszalin, 1977
[2] Fladrowska E.: Analiza czynników determinujących skuteczność wdrażania i stosowania systemu informatycznego w zarządzaniu przedsiębiorstwem przemysłowym. Rozprawa doktorska, Wydział Zarządzania UW, Warszawa, 1980
[3] Fladrowska E.: Czas systemów powtarzalnych. INFORMATYKA nr 2—3, 1982.



Rys. 2. Kontrola zwrotnego sygnalizowania błędów bazy danych podsystemu w sferze produkcji (materiały, części własne, miejsca wykonania operacji)
○ — punkt korekty błędów
TB — tabulogram
MNI — maszynowe nośniki informacji
TT — Dział Technologiczny
TK — Dział Konstrukcyjny
EP — Dział Planowania
EH — Dział Zbytu

WITOLD KLAUDEL
Instytut Chemii Nieorganicznej
Gliwice

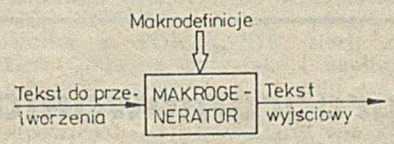
MC-82
makrogenerator o przeznaczeniu ogólnym

Konieczność przetwarzania tekstu występuje w wielu dziedzinach programowania. W takich zagadnieniach, jak edycje czy translacje wstępne [1] nieocenione usługi mogą oddać narzędzia zwane makrogeneratorami [1]. Makrogeneratory są programami [2], które dokonują przetwarzania tekstu na podstawie zadawanych reguł tzw. makrodefinicji (rys.). Każda makrodefinicja musi zawierać:

- **wzorzec makrowywołania (odwołania do makrodefinicji)**
- **definicję procesu rozwijania makrowywołania (tzw. treść makrodefinicji), będącą programem zapisanym w języku zleceń makrogeneratora.**

W poniższym artykule przedstawiony zostanie krótki opis makrogeneratora MC-82, zrealizowanego przez autora w języku PASCAL [2] na maszynie ODRA 1305 pracującej pod kontrolą systemu operacyjnego GEORGE-3. Przy projekcie MC-82 wykorzystano doświadczenia zdobyte podczas realizacji pracy [3].

MC-82 jest makrogeneratorem o przeznaczeniu ogólnym, nie związanym w zasadzie z żadnym językiem programowania. Jedynym uprzywilejowanym językiem jest FORTRAN, dla którego sporządzono specjalne zlecenie formatujące, a więc łagodzące wymagania pozycyjności tego języka.



Funkcja makrogeneratora

MC-82 pracuje w trybie swobodnym (tzn. makrowywołania nie muszą być w tekście specjalnie wyróżnione). Podstawową jednostką przetwarzania jest rekord tekstu mogący zawierać jeden lub kilka wierszy. Z MC-82 związane są dwa strumienie tekstu:

- strumień wejściowy, który zawiera inicjację oraz rekordy makrodefinicji przemieszane z rekordami makrowywołań i rekordami nie będącymi makrowywołaniami
- strumień wyjściowy, zawierający rekordy makrorozwinięć oraz nieprzetworzone rekordy z tekstu wejściowego.

[1] Tzn. przetwarzanie typu: program w języku niestandardowym → program w języku bazowym

[2] Istnieją realizacje, w których makrogeneratory stanowią integralną część translatora języka bazowego; tymi przypadkami jednak nie będziemy się zajmować

INICJACJA

Inicjacja zawiera opis warunków pracy makrogeneratora, co w tym przypadku oznacza przede wszystkim dobór znaków i symboli specjalnych, pozwalających prawidłowo interpretować pewne fragmenty tekstu. Dla wszystkich znaków i symboli przyjmuje się pewien standard, a inicjacja pozwala go zachować lub zmienić. Poniżej podamy kilka standardów niezbędnych dla dalszych rozważań:
÷ — znak początku komentarza w rekordzie treści makrodefinicji
& — znak początku zlecenia w rekordzie treści makrodefinicji
: — znak końca zlecenia w rekordzie treści makrodefinicji.

MAKRODEFINICJA I MAKROWYWOŁANIA

Każda makrodefinicja składa się z:

- rekordu otwierającego
- rekordu wzorca
- rekordów treści makrodefinicji, tzn. rekordów definiujących proces rozwijania makrowywołania
- rekordu zamykającego.

Na przykład — makrodefinicją może być:

MACDEF — standardowy rekord otwierający
WZÓR — rekord wzorca (przykładowy)
„treść" — rekordy treści makrodefinicji
MACEND — standardowy rekord zamykający.

Przyjmuje się prosty format dozwolonych makrowywołań, tzn. każde makrowywołanie składa się z:

- opcjonalnej etykiety, tj. napisu zaczynającego się od cyfry
- wzorca, tj. napisu zaczynającego się od litery
- parametrów rozdzielanych przecinkami.

Wzorzec musi być odseparowany od etykiety i od parametrów, co najmniej jedną spacją i w związku z tym ani etykieta, ani wzorzec nie mogą zawierać spacji.

Makrowywołaniem dla przykładowo podanej makrodefinicji mogą być np. rekordy:

WZOR 1,5
10A WZOR 2B, 3C

REKORDY TREŚCI MAKRODEFINICJI

Każdy rekord treści makrodefinicji składa się z części znaczącej i opcjonalnego komentarza, zapoczątkowanego specjalnym znakiem. Część znacząca składa się ze zleceń (otwieranych i zamykanych przez znaki specjalne) oraz dowolnych napisów nie zawierających znaku początku komentarza. Jeśli przyjmiemy standardowe znaki specjalne, to przykładowy rekord treści makrodefinicji może przyjąć postać:

& A1: = & A2: + & A3: ·/· SUMOWANIE

gdzie: & A1:, & A2:, & A3: — zlecenia; ·/. SUMOWANIE — komentarz.

OGÓLNY SCHEMAT DZIAŁANIA MC-82

Ważnym pojęciem związanym z MC-82 jest pojęcie BUFORA. Przez BUFOR przechodzą rekordy zarówno tekstu wejściowego, jak i wyjściowego, tu dokonuje się analizy ewentualnego makrowywołania i tu buduje się rekordy makrorozwinięć.

Spróbujmy teraz za pomocą uproszczonego i niekompletnego zapisu w języku PASCAL przedstawić schemat działania MC-82:

```
PROGRAM MAKROGENERATOR:
PROCEDURE PRZETWÓRZ; {rozpoznanie i przetworzenie
                      rekordu zawartego w BUFORZE}
VAR K,L,M:INTEGER;
 BEGIN
  IF NOT PUSTY (BUFOR) THEN
   BEGIN
    K: = ILOŚĆ MAKRODEFINICJI:
    WHILE NOT MAKRO (K) DO K: = K-1; {identyfikacja
                                      makrowywołania}
      PRZYGOTUJ (K); {przygotowanie do rozwijania
                      makrowywołania"K"}
    L: = 1, M: = ILOŚĆ REKORDÓW MAKRODEFINICJI [K-];
    WHILE L < = M DO
     BEGIN
      REALIZUJ REKORD (L); {przetworzenie rekordu treści
                            makrodefinicji}
      PRZETWÓRZ; {rekursja, umożliwiająca analizowanie
                  zagnieżdżeń, tzw. "makro w makro"}
      L: = L + 1
     END
   END
  END; {koniec PRZETWÓRZ}
BEGIN {początek programu głównego}
 INICJACJA;
 WHILE NOT EOF (INPUT) DO
  BEGIN
   DAJ BUFOR: {wprowadzenie rekordu tekstu wejściowego
                                  do BUFORA}
   PRZETWÓRZ
  END
END.
```

Analiza programu może prowadzić do przekonania, że nie jest w nim przewidziana obsługa wprowadzania makrodefinicji oraz sytuacji, w których BUFOR nie zawiera makrowywołania. Tak jednak nie jest. Dla przypadków tych zdefiniowano dwie wbudowane makrodefinicje:

- makrodefinicję o numerze kolejnym „2" (ze standardowym wzorcem MACDEF), zawierającą zlecenie tworzenia makrodefinicji
- makrodefinicję o numerze kolejnym „1" i pustym wzorcu (co powoduje, że każdy rekord może stać się jej makrowywołaniem), standardowo zawierającą zlecenie przesłania BUFORA do tekstu wyjściowego.

Kilka uwag wymaga etap przetwarzania rekordu treści makrodefinicji, a więc:

- przetwarzaniu podlega jedynie część znacząca rekordu
- napisy nie będące zleceniami przesyłane są do BUFORA
- zlecenia realizowane są przez specjalny procesor, a następnie ich treść jest pomijana.

MOŻLIWOŚCI LISTY ZLECEŃ MC-82

Za pomocą aktualnie dostępnej listy zleceń możliwe do wykonania są — przy użyciu MC-82 — takie operacje, jak:

- wprowadzanie makrodefinicji
- przesyłanie do BUFORA zmiennych tekstowych
- konwersje (lub przesłania) między zmiennymi dowolnych typów (z możliwością podejmowania akcji w przypadku błędów)
- wykorzystanie literałów do nadawania wartości zmiennym dowolnych typów
- operacje arytmetyczne, logiczne, konkatenacji na zmiennych i literałach odpowiednich typów (zapisywane w ONP)
- rozpoznawanie i wydzielanie łańcuchów znakowych w zmiennych tekstowych
- operacje na pomocniczych zbiorach tekstowych, tj. czytanie, zapisywanie, czyszczenie, testowanie końca, „przewijanie"
- testowanie i porównywanie zmiennych
- skoki, tj. zmiany w sekwencji przetwarzania rekordów treści makrodefinicji (do których odwołuje się przez ich kolejność występowania)
- przesyłanie BUFORA do strumienia tekstu wyjściowego (z ewentualnym formatowaniem FORTRANOWYM)
- czyszczenie BUFORA
- aktywacja i dezaktywacja wybranych makrodefinicji
- zmiana sposobu traktowania rekordów nie będących makrowywołaniami
- kopiowanie (bez analizowania) rekordów strumienia tekstu wejściowego do strumienia wyjściowego
- ignorowanie rekordów strumienia tekstu wejściowego.

LITERATURA

[1] Brown P. J.: Makrogeneratory i oprogramowanie przenośne. WNT, Warszawa, 1978

[2] Iglewski M., Madey J., Matwin S.: PASCAL. WNT, Warszawa, 1979

[3] Klaudel W.: Program makroasemblera dla minikomputera SMC-10. Praca magisterska, Politechnika Śląska, Wydział Automatyki i Informatyki, Gliwice, 1979.

Wielkie wydarzenia w historii oprogramowania

Zaczęło się od Augusty Ady Byron, urodzonej w 1815 roku. Zupełnie niepodobna do swego ojca, wielkiego angielskiego poety, pozostawiła po sobie spuściznę, którą dopiero teraz można właściwie ocenić. W ciągu krótkiego życia (zmarła mając 36 lat) udało jej się udokumentować wielkie dzieło Charlesa Babbage'a — maszynę różnicową. Napisała także program obliczania liczb Bernouliego, program, który czekał z górą 100 lat na swoje uruchomienie.

Ponad 90 lat później młoda matematyczka Grace M. Hopper została skierowana przez Marynarkę Stanów Zjednoczonych do pracy z Howardem Aikenem na uniwersytecie w Harvard. Wraz z początkiem pracy pani Hopper rozpoczęła się nowoczesna historia rozwoju oprogramowania. W ciągu niespełna 40 lat rozwinął się cały przemysł oprogramowania. Zatrudnienie wzrosło od jednego programisty do setek tysięcy osób, a inwestycje w oprogramowaniu są wręcz niemożliwe do oszacowania. W tym samym czasie opracowano setki języków programowania, wiele systemów operacyjnych i wiele programów do wszystkiego — od sterowania mikrofalami do obliczania orbit dla satelitów.

Mimo wszystko niewiele było wydarzeń, które określały kierunek rozwoju tego przemysłu. Wiele wydarzeń uważano za niezwykle ważne zaledwie przez bardzo krótki okres. Programowanie ma tę właściwość (której nie mają inne technologie), że wielkie przemiany są niemal natychmiast akceptowane.

Wiele dat zostało już zapomnianych. Ludzie wiedzą, że coś się wydarzyło w tym czy innym roku, ale mało kto pamięta dokładną datę. Nie ma dokładnej daty, co najwyżej można określić tydzień lub miesiąc, w którym program zaczął działać dostatecznie dobrze, liczba wdrożeń przewyższyła liczbę wykrytych błędów, a użytkownicy przestali narzekać na zawodność danego produktu.

Wydarzenia, które poniżej uwzględniono były arbitralnie wybrane przez autorów, będących Amerykanami i odzwierciedlają jedynie ich spojrzenie. Prawdopodobnie inni autorzy stworzyliby zupełnie inne zestawienie. Dla ułatwienia wydarzenia podzielono na dekady lat.

Lata czterdzieste i pięćdziesiąte: początek

1946 — Konrad Zuse nie mogąc zbudować odpowiedniego sprzętu przerzuca się na oprogramowanie i tworzy PLANKALKUL — język zawierający konstrukcje umożliwiające m.in. operowanie zmiennymi, które mogą reprezentować bardzo duże i bardzo małe liczby. Język ten można traktować jako pierwszy język algorytmiczny.

1949 — Francis Holberton wpada na pomysł generowania programu przez inny program; ideę tę wykorzystuje ona do generowania programów sortowania i łączenia.

1951 — dr Alec Glennie oddaje do eksploatacji kompilator AUTOCODE (dla komputera Ferranti MARK I), pierwszy kompilator dostosowany do ogólnego użytku.

1951 — Wilkes, Wheeler i Gill oraz — w tym samym czasie, zupełnie niezależnie — Grace Hopper wpadają na pomysł tworzenia i stosowania podprogramów.

1952 — Grace Hopper oddaje do eksploatacji kompilator A-0. W tym samym roku w dniu elekcji prezydenta USA, 4 listopada o godzinie 20.30, komputer UNIVAC I odniósł ogromny sukces, gdy po przeliczeniu zaledwie 7% głosów przewidział porażkę Eisenhowera. Przerażeni przedstawiciele rozgłośni CBS oraz firmy UNIVAC zredukowali oceny komputera ze 100:1 do 8:7, odbierając tym samym rozgłos dobrej analizie statystycznej oraz dobremu programowaniu.

1954 — John Backus dochodzi do wniosku, że język algorytmiczny mógłby być przydatny dla komputera IBM 704. Szef Backusa — Cuthburt Hurd aprobuje przedsięwzięcie.

1955 — w sierpniu Fletcher Jones zwołuje spotkanie potencjalnych użytkowników komputera IBM 704 z okręgu Los Angeles, uważając, że skoro firma IBM nie interesuje się oprogramowaniem, to może pomogą sobie wzajemnie użytkownicy. Jeszcze w tym samym miesiącu w czasie spotkania w siedzibie firmy RAND Corp. powstaje organizacja SHARE. Wraz z tą organizacją powstaje idea stworzenia standardów dla systemów operacyjnych, kompilatorów, translatorów, tablic połączeń itp.

1956 — IBM akceptując SHARE tworzy podobną organizację — GUIDE, ustanawiając tym samym istniejący do dziś podział.

1957 — w kwietniu uruchomiono w laboratorium firmy WESTINGHOUSE-BETTIS pierwszy program napisany przez użytkownika w FORTRANIE. Herb Bright, Ollie Swift oraz Lew Ondis z satysfakcją zaobserwowali, że kompilator „odrzucił" program z powodu braku nawiasu zamykającego. Pojawiło się wówczas pytanie, na które do dziś nie ma odpowiedzi: **jeśli kompilator wie jaki jest błąd, to dlaczego go nie naprawia?** Wraz z uruchomieniem tego programu rozpoczęła się era FORTRANU.

1958 — rozpoczyna swoje działanie system obrony powietrznej SAGE. Oprogramowanie obejmuje m.in. konwersacyjny dostęp do monitora graficznego, programowe „naprawianie" przekłamanych bitów, transmisję danych oraz symulatory otoczenia. Te ostatnie umożliwiły testowanie poszczególnych modułów — tak, jakby były częścią dużego systemu. Programiści zaangażowani we wczesnym stadium systemu SAGE czekają dotąd na pojawienie się takiego ogólnodostępnego rozwiązania. Firma SDC cierpiąc na brak programistów rozpoczyna szkolenie, które przez ponad siedem lat „produkuje" co dwa tygodnie 20 programistów.

Grupa naukowców niezadowolona z pewnych „komputerowych" ograniczeń FORTRANU uznaje za konieczne stworzenie nowego języka. Na serii spotkań zostaje stworzony International Algebraic Language, znany później jako ALGOL; jest to język starannie opracowany, o strukturze blokowej i rozwiniętych możliwościach wykonywania operacji matematycznych, całkowicie pomijający zagadnienia wprowadzania danych i wypisywania wyników.

1959 — Departament Obrony USA odrzuca stosowanie FORTRANU do realizacji swych zadań. Liczne komisje przystępują do stworzenia nowego języka, a całość prac koordynuje pani Grace Hopper. 25 lutego w MIT odbywa się pierwsza demonstracja języka APT (Automatically Programmed Tools). Język ten stanie się w późniejszych latach międzynarodowym standardem dla programowania obrabiarek sterowanych numerycznie. Doug Ross starannie przygotował prezentację tego języka: zebrani widzowie otrzymali popielniczki wyprodukowane przez sterowane numerycznie obrabiarki.

Lata sześćdziesiąte: eksplozja oprogramowania

1960 — na spotkaniu, które odbyło się w Paryżu podjęto wiążące decyzje o ALGOLU oraz opracowano raport, który do dziś służy jako wzór definicji języka. Ale język ALGOL jest stosowany niemal wyłącznie przez stosunkowo niewielką grupę europejskich naukowców.

W lutym tego roku zostaje oddany do użytku język MAD (Michigan Algorythmic Decoder); MAD jest oparty na ALGOLU 58 i jest wynikiem działania trzech osób: Bruce Ardena, Boba Grahama i Bernie Galler. MAD zawiera specjalne konstrukcje umożliwiające łatwe definiowanie nowych instrukcji.

Także w tym samym roku zostaje uruchomiony w firmie SDC język JOVIAL — Jule's (Schwartz) Own Version of the International Algebraic Language. Dla setek programistów oznacza to ostateczne pożegnanie z językiem typu asemblera.

W grudniu tegoż roku napisany w COBOLU program zostaje wykonany na komputerach UNIVAC II i RCA 501, a otrzymane wyniki są identyczne. Twórcami tego osiągnięcia byli Charlie Phillips, Howard Bromberg i Grace Hopper. Wszyscy producenci komputerów zaczynają spiesznie przygotowywać własne translatory COBOLU. Podobne zachowanie niestety już nigdy się nie powtórzy, rozpoczęła się bowiem ochrona oprogramowania.

Kierowany przez Boba Burtona, mały zespół w firmie BORROUGHS, zakończył właśnie swe dzieło — komputer B-5500 oparty na zastosowaniu ALGOLU. Burton forsował dziwną ideę — zaprojektowanie komputera dostosowanego do oprogramowania. Dla ułatwienia kompilacji i wykonania zadań stosowane są specjalne instrukcje obsługi stosu. W owym czasie mało osób wierzyło w to przedsięwzięcie, a i obecnie niewielu (z godnymi odnotowania wyjątkami) w to wierzy.

1961 — Maurice Halstead, pracujący na Uniwersytecie w Berkeley oddaje do użytku kompilator (opartego na koncepcji ALGOLU) języka NELIAC. Ciekawą właściwością tego języka jest to, że jego translator został napisany również w języku NELIAC. Uważa się, że jest to pierwszy kompilator napisany we własnym języku.

1962 — John McCarthy, w rezultacie sześcioletnich prac, oddaje do użytku LISP — język do zagadnień związanych ze sztuczną inteligencją i operowaniem symbolami. Istotnym osiągnięciem jest tu rozwiązanie zagadnienia wykorzystywania zwalnianej pamięci, zagadnienia, które dziś już praktycznie nie istnieje.

1963 — w listopadzie, laboratorium naukowe firmy GENERAL MOTORS kończy sześcioletni okres, w którym po raz pierwszy zaprojektowano część samochodu korzystając z terminala dołączonego do komputera. System DAC-1 zawiera własny system zarządzania bazą danych, języki programowania, możliwość koordynacji pracy wielu procesorów, wirtualną pamięć do przechowywania danych oraz możliwość szybkiego wydrukowania obrazu wyświetlonego na ekranie terminala. Ponieważ GENERAL MOTORS nie jest zainteresowany uzyskiwaniem stopni naukowych, sukcesu tego nie rozgłoszono.

W tym samym roku firma GENERAL ELECTRIC oddaje do użytku, opracowany przez niewielki zespół pod kierownictwem Charlie Bachmana system IDS (Integrated Dats Store), pierwszy system zarządzania bazą danych. Ze względu na małą liczbę użytkowników komputera GE 225 (dla którego zrobiono ten system) IDS przez długi czas nie zdobył większego rozgłosu. W późniejszym okresie opracowano wiele nowych wersji IDS, który stał się pierwowzorem dla dzisiejszych systemów zarządzania bazą danych. Trudno było przewidzieć, że system opracowany dla jednego klienta może znaleźć tak szerokie pole zastosowań.

1964 — John Kemeny i Tom Kurtz z Dartmouth College oddają do użytku BASIC — język, który został szczególnie szeroko rozpowszechniony. Celem twórców było ułatwienie korzystania z komputera tym użytkownikom, którzy nie są programistami. BASIC nie od razu został zaakceptowany, a mało kto wierzył, że jest to język oparty na ALGOLU. Opis pierwszej wersji języka oraz podręcznik użytkownika zajmowały kartkę o wymiarach 76×127 mm.

W kwietniu IBM ogłasza wprowadzenie na rynek komputerów SYSTEM/360. Według oświadczenia kierownictwa firmy, wydawano ponad 50 mln dol. rocznie i zużyto 5000 osobo-lat na opracowanie systemu operacyjnego OS/360.

W grudniu IBM wydaje opis nowego języka NPL. Ze względu na zastrzeżenia National Physical Laboratory (również skrót NPL!) nazwę tego języka zmieniono na PL/I. Był to jeden z ostatnich projektów, w których użytkownicy grali główną rolę. IBM przejął kontrolę nad rozwojem tego języka, a główni projektanci z ramienia SHARE — Bruce Rosenblatt, Hans Burg oraz Jim Cox — zostali przytłoczeni wymaganymi przez IBM „udoskonaleniami".

1965 — Larry Breed i Adin Falkoff rozpoczynają implementację opracowanego przez Kena Iversona języka APL. Język ten, opracowany początkowo przez Freda Brooksa jeszcze w 1956 roku, był przeznaczony do definiowania i opisywania różnych funkcji komputera. Implementacja trwała cztery lata, ale uzyskano zwarty i dobrze zdefiniowany język o dużych możliwościach.

Holenderski profesor Edsger W. Dijkstra podejmuje problem tzw. zakleszczenia, który występuje np. jeśli dwa konkurujące ze sobą programy czekają na te same zasoby. Jego prace doprowadziły do wielu późniejszych rozwiązań; warto jednak podkreślić, że do czasu publikacji Dijkstry wspomniany problem był zupełnie niezrozumiały. Profesor Dijkstra opracował później system operacyjny THE, który doskonale rozwiązywał zagadnienie przydziału zasobów.

1968 — w październiku odbywa się w Garmisch Partenkirchen (RFN) pierwsza z dwóch organizowanych przez NATO konferencji, w której bierze udział wielu sławnych informatyków. Konferencja dotyczy różnych zagadnień oprogramowania, włącznie z tak nienaukowymi, jak cena oprogramowania.

Profesor Edsger Dijkstra pisze do redakcji czasopisma Communications of the ACM list, w którym stwierdza, że **„jakość programisty jest odwrotnie proporcjonalna do liczby instrukcji GOTO użytych w jego programach. Ostatnio odkryłem, dlaczego użycie instrukcji GOTO ma tak fatalne skutki i uświadomiłem sobie, że instrukcja ta powinna być usunięta ze wszystkich języków wysokiego poziomu".** Tym samym Dijkstra postawił zagadnienie programowania strukturalnego.

1969 — 23 czerwca IBM ogłasza, że od tego dnia oprogramowanie będzie sprzedawane. Choć oświadczenie to ma niewielki wpływ na bieżące wydatki użytkowników, to jednak w ciągu 10 lat IBM stworzyło olbrzymi przemysł o potencjale szacowanym na 3×10^{12} dol.

W tym samym roku oddano do eksploatacji system abonencki MULTICS, opracowany w MIT dla komputerów GENERAL ELECTRIC. System ten zastąpił swego ograniczonego w możliwościach poprzednika CTSS (Compatible Time Sharing System), który działał na komputerach IBM 7094. MULTICS służył do wyszkolenia wielu studentów MIT. Ciekawostką jest to, że MULTICS został napisany w języku PL/I.

Lata siedemdziesiąte: reguły programowania

1970 — zostaje opracowany amerykański standard języka COBOL, jednak mimo tego poszczególni sprzedawcy translatorów tego języka rozwijają własne wersje. Opracowany standard nie ma praktycznego zastosowania, to samo zresztą powtórzy się w roku 1981, kiedy zostanie opracowana następna wersja tego standardu.

1971 — zostaje wydana książka Gerry Weinberga „The Psychology of Computer Programming". Weinberg wzywa do programowania strukturalnego, stosowania języków wysokiego poziomu oraz tworzenia zespołów programistów, a więc postuluje wszystko to, czego programiści bardzo nie lubią. Książka tak dobrze wyjaśnia zagadnienia programowania, że przez najbliższych 10 lat jest traktowana jako podręcznik.

IBM ogłasza opracowanie pamięci wirtualnej, dzięki czemu programiści będą zwolnieni od potrzeby optymalizacji wykorzystania pamięci oraz segmentacji programów. Wkrótce jednak wiele ośrodków stwierdza, że aby wykorzystać to osiągnięcie trzeba kupić dodatkową pamięć. Ponadto wkrótce potem okazuje się, że idea ta była już w 1958 roku zrealizowana w Anglii dla komputera ATLAS oraz w 1961 roku w USA — dla B-5500.

1975 — Fred Brooks, współtwórca języka APL oraz kierujący realizacją systemu operacyjnego OS/360 wydaje książkę „The Mythical Man-Month", w której szczegółowo analizuje tradycyjną filozofię programowania. Jest to druga książka o tak dużej wartości z punktu widzenia oprogramowania.

Departament Obrony Stanów Zjednoczonych zaczyna myśleć o oprogramowaniu tysięcy komputerów umieszczonych w różnych rodzajach broni. Rozpisany zostaje konkurs, który wygrywa firma HONEYWELL-BULL. Ostateczny rezultat — język ADA, realizowany pod kierunkiem Jean Ichbiah, jest jednolitym i zwartym produktem.

1978 — w czerwcu odbywa się w Los Angeles konferencja zorganizowana przez firmę SIGPLAN na temat historii języków programowania.

1979 — popularny staje się VISICALC oraz następuje eksplozja minikomputerów. VISICALC umożliwia łatwe stosowanie informatyki w przedsiębiorstwie bez konieczności nauki programowania. Umożliwia to niespotykany dotąd rozwój sprzedaży komputerów. Początkowe rozwiązanie oparte było na mikrokomputerze APPLE, ale wkrótce VISICALC rozszerzył je na inne mikrokomputery. Można stwierdzić, że to osiągnięcie jest tej miary, co wprowadzenie FORTRANU, ponieważ udostępniło komputery ludziom bez specjalnego wykształcenia technicznego.

Lata osiemdziesiąte: oczekiwania

1980 — IBM dostarcza System/38, pierwszy system komputerowy z wbudowanym relacyjnym systemem zarządzania bazą danych. Zostaje wreszcie zrealizowana, opracowana 10 lat wcześniej idea Codda. Trudno sobie jednak wyobrazić praktyczną konwersję olbrzymiej liczby rekordów z baz danych zarządzanych przez system IMS do baz relacyjnych.

1981 — INTEL ogłasza wprowadzenie mikroprocesora 432, programowanego całkowicie w języku ADA. Ponadto firma SOFTWARE AG ogłasza, że będzie sprzedawała system zarządzania bazą danych zrealizowany sprzętowo, dostosowany do możliwości istniejących komputerów.

* * *

Oprogramowanie stało się przemysłem, na którym się zarabia i traci pieniądze. Zadziwiająca jest różnorodność oferowanych programów, niektóre są sprzedawane wraz ze sprzętem, inne — oddzielnie. To co kiedyś było zajęciem dla pojedynczych osób, stało się planową działalnością wielkich przedsiębiorstw. Ciekawe jest, że dwa wielkie osiągnięcia — FORTRAN i VISICALC — zostały zrealizowane odpowiednio przez 13 i 2 osoby. Może to oznacza, że oprogramowanie lat osiemdziesiątych będzie należało do małych firm?

Co się stało z pionierami oprogramowania, którzy wyznaczali jego historię? Niektórzy umarli, ale wielu jest nadal aktywnych: Grace Hopper już dawno przekroczyła wiek emerytalny, ale ciągle szkoli nowe generacje programistów, John Backus ciągle pracuje w IBM, Doug Ross ciągle automatyzuje projektowanie, Iverson, Falkoff i Breed nadal pracują nad APL, a Herb Bright nad oprogramowaniem dla celów kryptografii.

Opracował JAN WIERZBOWSKI
na podstawie artykułu
„Magic Moments in Software"
Datamation, sierpień 1981

Generator zmiennej losowej o rozkładzie normalnym

Przedstawiamy poniżej procedurę realizującą generator zmiennej losowej o rozkładzie normalnym na minikomputerze MERA 400. Sądzimy, że spotka się ona z zainteresowaniem użytkowników BASICA i znajdzie zastosowanie praktyczne — szczególnie w problematyce rolniczej, gdzie procesy produkcji zależą od wielu różnych czynników, zwłaszcza warunków przyrodniczo-klimatycznych.

MOTYWACJA

Przedstawiony algorytm generowania liczb losowych o rozkładzie normalnym może być istotną częścią każdego programu badającego właściwości procesów przyrodniczych, technicznych i ekonomicznych w większych przedsiębiorstwach, typu kombinatów państwowych gospodarstw rolnych.

Omawiany generator zmiennych losowych może też być bardzo użytecznym narzędziem — z praktycznego punktu widzenia — w statystyce doświadczeń rolniczych, ponieważ problem opisu określonych populacji za pomocą próbek losowych z tych populacji jest typowym problemem w badaniach rolniczych.

OPIS ROZWIĄZANIA

W rozwiązaniu zastosowano metodę eliminacji i superpozycji rozkładów (por. R. Zieliński: Generatory liczb losowych. WNT, Warszawa, 1979, s. 85—89).

Podana w poniższym artykule procedura generuje zmienną losową **X** o rozkładzie normalnym **N (0, 1)**:

- z prawdopodobieństwem 0,8638 jako zmienną losową określoną wzorem:

$$X = 2\left(R_1 + R_2 + R_3 - \frac{3}{2}\right)$$

gdzie R_1, R_2 i R_3 są niezależnymi zmiennymi losowymi o rozkładzie równomiernym w przedziale (0, 1)

- z prawdopodobieństwem 0,9745 jako zmienną określoną wzorem:

$$X = 3\,(R_1 + R_2 - 1)/2$$

- z prawdopodobieństwem 0,9973002039 jako zmienną losową o rozkładzie z gęstością

$f_3(x)/0{,}0228002039$

przy czym

$$f_3x = \begin{cases} c_1 e^{-x^2/2} - c_2\,(3 - x^2) - c_3\,(1{,}5 - |x|) & \text{dla } |x| \leqslant 1 \\ c_1 e^{-x^2/2} - c_4\,(3 - |x|)^2 - c_3\,(1{,}5 - |x|) & \text{dla } 1 < |x| \leqslant 1{,}5 \\ c_1 e^{-x^2/2} - c_4\,(3 - |x|)^2 & \text{dla } 1{,}5 < |x| \leqslant 3 \\ 0 & \text{dla } |x| > 3 \end{cases}$$

gdzie $c_1 = 17{,}49731196$; $c_2 = 4{,}73570326$; $c_3 = 2{,}15787544$; $c_4 = 2{,}36785163$.
W procedurze oznaczono ją symbolem **F3**. Jest ona skupiona na przedziale (-3,3), przy czym:

$f_3(x) < 0{,}358$.

Za wygenerowaną wartość z „ogona" $(3, +\infty)$ rozkładu normalnego przyjęto liczbę $(9\text{-}2 \ln R_1)^{1/2}$ (por. Marsaglia G.: Generation a variable from the tail of the normal distribution. Technometrics, 6, 1964, s. 101—102).

W przypadku, gdy należy generować zmienną **X1** o rozkładzie normalnym **N (M, S1)**, generujemy **X** zgodnie z rozkładem **N (0, 1)** i obliczamy **X1** według wzoru:

$X1 = S1 \cdot X + M$

gdzie **S1** — odchylenie standardowe, **M** — wartość średnia.

Procedura wczytuje wartości **M, S1, K** i wypisuje **K** liczb o rozkładzie N(M, S1).

Algorytm realizujemy za pomocą procedury GENERATOR. Czas wykonania omawianej procedury dla **K** = 100 wynosi ok. 30 sekund.

```
10 REM GENERATOR
20 PRINT "PODAJ WARTOSC SREDNIA M I ODCH. STAND. S1"
30 INPUT M, S1
40 PRINT "PODAJ ZADANA ILOSC LICZB O ROZKLADZIE
                                        N (M, S1)"
45 INPUT K
50 FOR I=1 TO K
60 R=RND
70 IF R<0.8638 GO TO 320
80 IF R<0.9745 GO TO 370
90 IF R<0.9973002039 GO TO 160
100 R1=RND
110 R2=RND
115 REM LOG — LOGARYTM NATURALNY
120 IF R2 <=(3/SQR (9—2*LOG (R1)) GO TO 140
130 GO TO 100
140 X=SQR (G—2* (LOG (R1))
150 GO TO 400
160 R1=RND
170 R2=RND
180 R1=6*R1 — 3
190 R2=0.358*R2
200 IF ABS (R1) =1 GO TO 250
210 IF ABS (R1) <=1.5 GO TO 270
220 IF ABS (R1) <= 3 GO TO 280
230 F3=0
240 GO TO 290
250 F3=17.49731196*EXP (- (R1*R1) /2) - 4.73570326*(3-R1*R1)
251 F3=F3-2.15787544*(1.5-ABS (R1))
260 GO TO 290
270 F3=17.49731196*EXP (- (R1*R1) /2) -2.36785163*
                        (3-ABS (R1))*(3-ABS (R1))
271 F3=F3-2.15787544*(1.5-ABS (R1))
275 GO TO 290
280 F3=17.49731196*EXP (- (R1*R1)/2)-2.36785163*
                        (3-ABS (R1))*(3-ABS (R1))
290 IF R2>F3 GO TO 160
300 X=R1
310 GO TO 400
320 R1=RND
330 R2=RND
340 R3=RND
350 X=2*(R1+R2+R3 - 3/2)
360 GO TO 400
370 R1=RND
380 R=RND
390 X=3/2*(R1+R2—1)
400 X1=X*S1+M
410 PRINT"LICZBA O ROZKLADZIE N (0, 1) = ";X
411 PRINT"LICZBA O ROZKLADZIE N (M, S1) = ";X1
420 NEXT I
430 END
```

JADWIGA ORYLSKA
ANTONI MIKLEWSKI

Generowanie par liczb z rozkładem normalnym

Poniżej zaprezentowana procedura generuje parę niezależnych liczb losowych o rozkładzie normalnym, z wartością średnią **M** i odchyleniem standardowym **S**. Może ona znaleźć zastosowanie w programach symulacyjnych, grach, obliczeniach numerycznych itd. Wykorzystana tu została metoda Boxa-Marsaglii-Mullera, znana również jako „metoda współrzędnych biegunowych" (por. Knuth D. E., The Art of Computer Programming, vol. 2 — Seminumerical Algorithms).

Zakładamy, że funkcja standardowa **RND(0)** generuje liczby pseudolosowe z rozkładem równomiernym na przedziale (0, 1), **LOG(X)** oznacza logarytm naturalny z **X**, zaś wynik reprezentowany jest przez zmienne **N1, N2**. A oto treść procedury:

```
100 REM WYNIKIEM DZIAŁANIA GENERATORA
      JEST PARA LICZB PSEUDOLOSOWYCH
110 REM N1 I N2 (ROZKLAD NORMALNY N (M, S))
120 P = 3.141593
130 R1 = RND(0)
140 R2 = 2*P*RND(0)
150 REM 0<R1<1, 0<R2<2*P
160 R = SQR(-2*LOG(R1))
170 T1 = R*SIN(R2)
180 T2 = R*COS(R2)
190 REM T1, T2 SA LICZBAMI PSEUDOLOSOWYMI
          O ROZKLADZIE NORMALNYM N(0, 1)
200 N1 = S*T1 + M
210 N2 = S*T2 + M
220 RETURN
```

JAKUB TATARKIEWICZ

PRINTEMPS INFORMATIQUE czyli informatyczna wiosna

Pierwsze dni wiosny wywołały ożywienie nie tylko w przyrodzie, lecz także w światku informatycznym. W dniach 22—25 marca, już po raz dziewiąty w Paryżu odbył się salon Printemps Informatique (Informatyczna Wiosna). Jak wykazały podsumowania statystyków — wzięło w nim udział 182 wystawców i 9700 zwiedzających, w tej liczbie niżej podpisani.

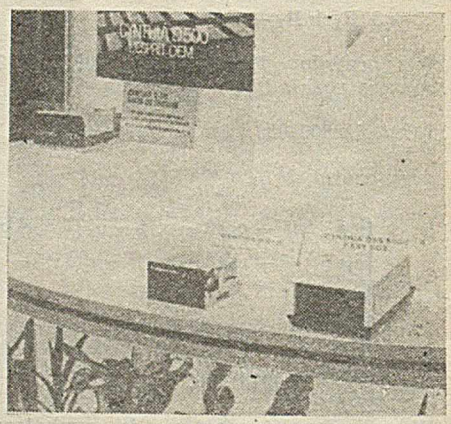
Gdy w 1975 roku spotykano się po raz pierwszy, był to salon przeznaczony wyłącznie dla importerów amerykańskiego sprzętu, a zorganizowany przez izbę handlu USA. Dopiero piąta edycja (1979), wraz ze zmianą organizatora, przyniosła udział producentów europejskich. Natomiast w tym roku zmieniono podtytuł wystawy: z „Międzynarodowe Forum Informatyki" na „Zawodowe Forum Informatyki OEM". Chciano w ten sposób podkreślić jej roboczy charakter — bo w istocie nie jest to SICOB adresowany do szerokiej publiczności, ale impreza raczej typu targowego, przeznaczona przede wszystkim dla reprezentantów SSCI (informatyczne przedsiębiorstwa usługowo-doradcze), przez które najczęściej sprzęt trafia ostatecznie do użytkownika. Dlatego też wstęp na Printemps Informatique jest bezpłatny, choć w zasadzie należy posiadać zaproszenie. Piszemy „w zasadzie", gdyż wystarczyło nam wypełnienie kart informacyjnych przy wejściu i już po chwili, z przypiętymi znaczkami uczestników, chodziliśmy między stoiskami.

Wszechświatowa inwazja mikroelektroniki zmienia oblicze tego typu salonów OEM — i to nie tylko dlatego, że większość oferowanych drukarek, jednostek dyskowych czy monitorów ekranowych była wykonana w nowo obowiązujących standardach, ale także dlatego, że pojawia się na nich niemało gotowych mikrokomputerów. I Paryż nie był pod tym względem wyjątkiem. Zresztą, jak wykazała ankieta, 48% zwiedzających było zainteresowanych tą właśnie grupą wyrobów.

Ale skoro Printemps Informatique ma w swym tytule OEM — zacznijmy od urządzeń peryferyjnych.

Pamięci magnetyczne

Równolegle ze wzrostem pojemności i spadkiem ceny, urządzenia peryferyjne zmierzały w kierunku wyznaczanym przez rozwój jednostek centralnych, systematycznie zmniejszając swoje wymiary. Dyski sztywne, które jeszcze niedawno istniały wyłącznie jako 14-calowe, zmniejszyły swoje rozmiary do 8 cali, a następnie do 5,25 (wraz z wprowadzeniem technologii Winchester). Te ostatnie wymiary pojawiły się po raz pierwszy na światowych wystawach w roku 1980 — dziś, po trzech zaledwie latach, można wybierać spośród kilkudziesięciu modeli pochodzących od kilkunastu producentów. Gdy rozpoczynano produkcję, wszystkich szokowała ich pojemność — aż 5 M bajtów. W tym roku pokazano modele o pojemności 40 MB (COMPUTER MEMORIES), 46 MB (ATASI), a nawet ponad 50 MB (EVOTEK).



Stoisko CII HB; to maleństwo na podwyższeniu, to pamięć dyskowa CYNTHIA D520 (26 Mbajtów)

Zmieniają się też walory użytkowe tych urządzeń — o ile pierwsze konstrukcje obejmowały wyłącznie dyski

stałe, o tyle obecnie można wybrać z kilku modeli, w których oprócz dysku stałego jest również kaseta z dyskiem wymiennym. Przykładem takiego rozwiązania był model CYNTHIA D520 firmy CII HONEYWELL BULL (pojemność: część stała — 13 MB, część wymienna — 13 MB). Warto tu przypomnieć, że cały rozwój jednostek dyskowych 5,25 cala odbywa się przy niezmienianych, standardowych wymiarach zewnętrznych, np. wysokość wynosi 82,6 mm.

Zamiast zwyczajowego podawania gęstości zapisu, czasu dostępu do ścieżki czy szybkości transmisji, chcielibyśmy jednak zwrócić uwagę na inny parametr tych urządzeń — średni czas międzyawaryjny (MTBF). Otóż obecnie, jeśli wprowadza się jakiś wyrób na rynek, to minimalny MTBF wynosi 8 tys. godzin, zaś przeciętny — 10 tys. (przypominamy, że rok ma średnio 8756 godzin). Można tu jeszcze dodać, że średnia awaria trwa 60 minut.

Chociaż urządzeń tego typu jest jeszcze bardzo mało, wszyscy już mówią o pamięciach dyskowych o średnicy nośnika poniżej 4 cali (czyli poniżej 10 cm). Wiadomo już, że w najbliższym czasie rynek powinien rozstrzygnąć, jaki standard zwycięży — 3, 3,25 czy 3,5 cala. Na razie na Printemps Informatique jedyną pełną ofertą handlową były jednostki dyskietek 3-calowych firmy HITACHI. Ale wiadomo już, że przodownik tego rynku — CONTROL DATA CORP. — pokaże nieco później (w tym roku) swoje nowe konstrukcje 3,5-calowe: dyski sztywne CRICKET i elastyczne JIMINY. Jednak nad całą tą walką unosi się cień giganta — IBM pracuje nad dyskami średnicy 3,9 cala! Oczywiście rozwój mikrodysków związany jest z coraz częstszym pojawianiem się mikrokomputerów przenośnych, a umożliwia go postęp technologiczny w gęstości zapisu (metalowe nośniki magnetyczne, zapis pionowy i głowice cienkowarstwowe).

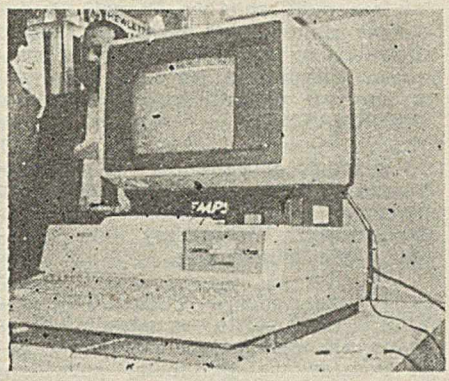
Drukarki

Mikroinformatyka nie mogła także nie wywrzeć wpływu na rozwój drukarek — niestety tego, co było najbardziej charakterystyczne, nie daje się tu odtworzyć — jakość druku tych urządzeń jest o wiele wyższa niż naszego miesięcznika. Druga podstawowa cecha także wymaga uwierzenia na słowo — nowoczesne drukarki są bardzo ciche (np. przy szybkości 600 wierszy na minutę poziom natężenia dźwięku nie przekracza 50 dBA).

Zatem trochę ciekawostek. CENTRONICS pokazał specjalny podajnik do swojej drukarki 353, który pozwala wykorzystywać zwykły papier maszynowy (w arkuszach) oraz koperty — nic dziwnego, skoro w wielu zastosowaniach wyznacznikiem poziomu nowszych konstrukcji jest tzw. jakość listu. Natomiast QUME zaprezentował model Sprint 11/40, który ma wymienne kasety interfejsu — do wyboru V24, IEEE, IBM PC, QUME i CENTRONICS.

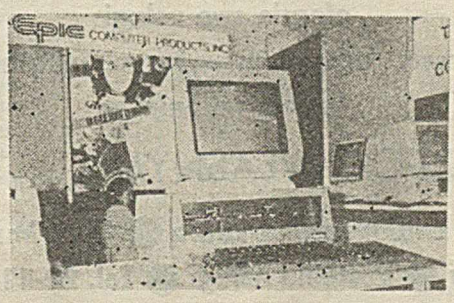
Mikrokomputery

Ponieważ Printemps Informatique to impreza przeznaczona dla zawodowców, prezentowane na niej mikrokomputery pochodziły prawie wyłącznie z grupy konstrukcji profesjonalnych, co oznacza, że były to maszyny co najmniej 16-bitowe. Co najmniej, gdyż w stoisku firmy HEWLETT-PACKARD królował, opisywany już na naszych łamach, 32-bitowy HP 9000.

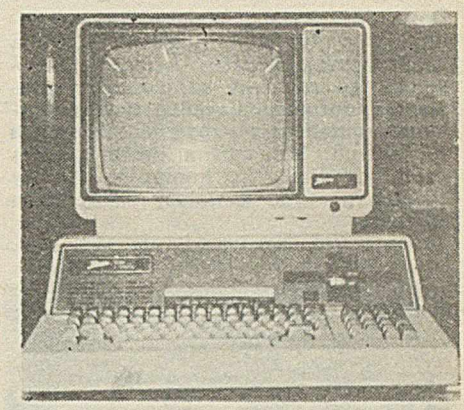


32-bitowy stołowy mikrokomputer HP 9000

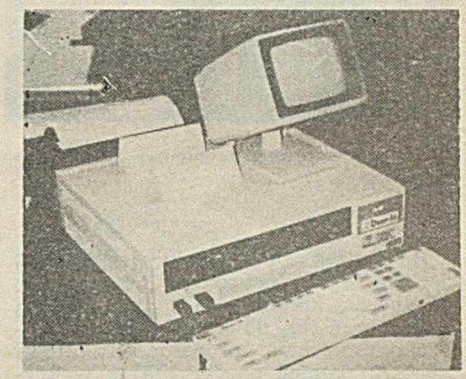
Jednak najliczniej reprezentowane były maszyny 16-bitowe — można powiedzieć, że istnieją już pewne standardy tego sprzętu. Mianowicie: system operacyjny MS-DOS lub CP/M, pamięć operacyjna przynajmniej 128 K bajtów (z możliwością rozbudowy ponad 0,5 MB), 1—2 jednostki pamięci na dyskach elastycznych, interfejs dysku sztywnego i cena od 30 do 50 tys. franków za konfigurację podstawową. W tej klasie pokazano znane już wcześniej konstrukcje, np. IBM PERSONAL COMPUTER, VICTOR S1 (dawny SIRIUS), ZENIT Z-100 czy DEC RAINBOW 100 (te dwie ostatnie mają po dwa procesory — 8- i 16-bitowy), a także nowości wystawy: francuski SIL'Z 16 (LEONARD) i japoński SBC Duet-16 (SAKATA SHOKAI). Można też mówić o stabilizacji cen mikrosystemów wieloterminalowych (np. ALTOS, AI-M16, MICROMULTI) na poziomie 60—150 tys. FF.



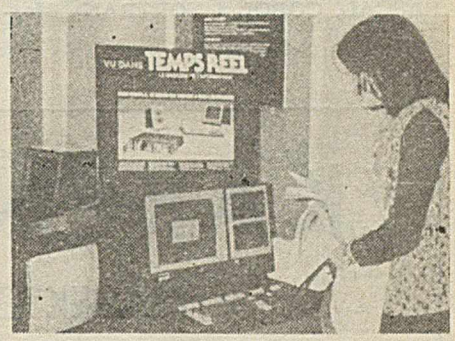
Wysokie walory użytkowe, umiarkowana cena i zgodność z IBM PC — czyli VICTOR S1



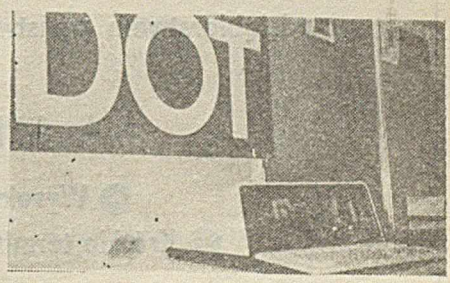
Duża moc obliczeniowa procesora 16-bitowego i możliwość wykorzystania bogatego oprogramowania maszyn 8-bitowych: dwuprocesorowy ZENITH Z-100



Jedna z niewielu premier — SBC Duet-16

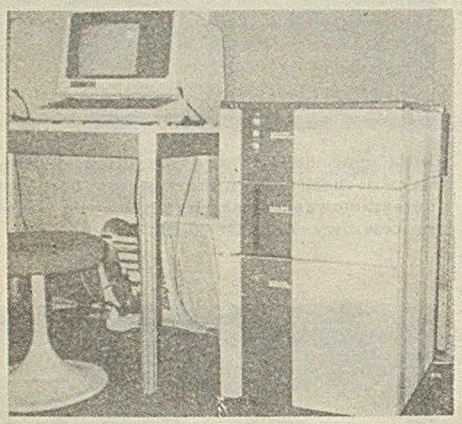


Autorka relacji i komputer-kameleon SMT GOUPIL 3



COMPUTER DEVICES DOT, czyli według reklamy — „przenośna moc obliczeniowa"

Do żadnej kategorii nie daje się natomiast zaliczyć komputera GOUPIL 3, lansowanego przez firmę SMT — można go zamówić w jednej z dziesięciu konfiguracji, dobierając dodatkowe urządzenia spośród dziesięciu możliwych. Nie można nawet powiedzieć, czy jest to komputer 8- czy 16-bitowy — bo to zależy od życzenia klienta.



Modułowy supermikrokomputer ZILOG System 8000

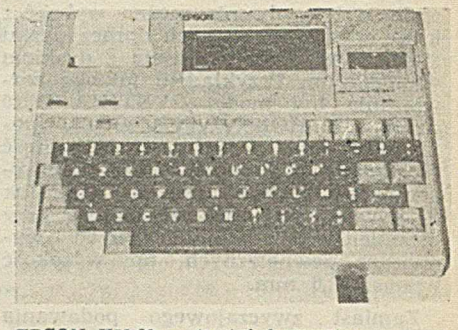
Również klasą dla siebie jest ZILOG System 8000 pochodzący od znanego producenta układów scalonych. Można go zaliczyć tak do mini- jak i do mikrokomputerów: procesor 16-bitowy Z8001A, pamięć operacyjna (z wykrywaniem i poprawianiem błędów) od 1 do 4 MB, moduł pamięci dyskowej 84 MB, system operacyjny UNIX i translatory siedmiu języków (w tym PASCAL, COBOL i FORTRAN). Można również dokupić jednostkę pamięci taśmowej o gęstości zapisu 6400 bpi. Modułowa konstrukcja pozwala w szerokim zakresie rozbudowywać ten komputer — w przyszłości przewidziano nawet dołączanie jednostek 32-bitowych.

Największą sensację wśród zwiedzających wzbudził jednak ostatni przebój firmy APPLE, czyli — LISA. Przejawiło się to w stałym oblężeniu stoiska, do którego trudno było się przecisnąć. I nie można się temu dziwić — wszak LISA wystąpiła pierwszy raz publicznie we Francji. A ponieważ stanowi ona pewnego rodzaju rewolucję w dziedzinie mikrokomputerów, poświęcimy jej osobny artykuł w jednym z najbliższych numerów INFORMATYKI.

Nie mogło zabraknąć na tym prestiżowym salonie reprezentantów nowo tworzącej się grupy mikrokomputerów przenośnych. Wybraliśmy jako przykład dwie konstrukcje.

Pierwsza z nich — COMPUTER DEVICES DOT — to urządzenie przenośne, ale wymagające zasilania z sieci. Podstawowe parametry pozwalają uznać je za sprzęt profesjonalny — 64—704 KB pamięci RAM, 64 KB ROM, procesor 16-bitowy INTEL 8088 z możliwością dołączenia procesora arytmetycznego INTEL 8087, dwie jednostki pamięci dyskietkowej 3,5 cala (ponad 400 KB każda), ekran o przekątnej 9 cali (1056×248 punktów), drukarka termiczna oraz do wyboru kilka interfejsów komunikacyjnych. Wszystko to razem waży 13 kg.

Natomiast EPSON HX-20, to urządzenie całkowicie przenośne, o wadze 1,7 kg i z własnym zasilaniem bate-



EPSON HX-20 — to też komputer

ryjnym (50 godzin pracy). Zawiera dwa procesory 8-bitowe, 16—32 KB RAM, 32 KB ROM, mikrokasetę (2×50 KB), ekran — 4 wiersze po 20 znaków (ciekłe kryształy), drukarkę (graficzną!) o szerokości 24 znaki (144 punkty) w wierszu oraz standardowe interfejsy: RS-232C, komunikacyjny, magnetofonu kasetowego oraz czytnika kodów paskowych. W dodatkowym wyposażeniu — interfejsy do telewizora i dyskietek oraz modem akustyczny.

* * *

Tegoroczny salon Printemps Informatique nie przyniósł żadnych szokujących nowości — jedno z czasopism nawet napisało, że był szary. Nie zmieniła się również — w stosunku do roku poprzedniego — liczba zwiedzających. Ale może ta szarość oznacza, że mikroinformatyka wtopiła się w codzienność, że przestała być tylko ciekawostką.

MARIANNA i MAREK SOBCZYKOWIE
(zdjęcia autorów)

„KAREN"

Przedsiębiorstwo Zagraniczne w Polsce
Warszawa, ul. Krochmalna 2 m. 127

PROJEKTOWANIE I WDRAŻANIE SYSTEMÓW INFORMATYCZNYCH, PRZETWARZANIE DANYCH

oferuje swe usługi:

przedsiębiorstwom państwowym, uczelniom wyższym, instytucjom badawczo-naukowym, firmom polonijnym

NA SPRZĘCIE WŁASNYM LUB KLIENTA

● Wysoko kwalifikowana kadra ● Ceny konkurencyjne
● Krótkie terminy ● Rzetelne usługi ● Tel. 20-11-42 oraz 17-43-91

EO/861/K/83

USA, Europa Zachodnia, Japonia

Rynek elektroniczny

Co roku dwutygodnik ELECTRONICS zamieszcza w swym pierwszym numerze wyniki ankiety przeprowadzonej wśród zachodnich firm elektronicznych. Obejmuje ona analizę produkcji w minionym okresie oraz przewidywania na najbliższe lata. Ponieważ ankieta przeprowadzana jest w końcu roku, kiedy wyniki ostatniego kwartału nie są jeszcze znane, dane za analizowany rok (tu: 1982) są przewidywane na podstawie trzech pierwszych kwartałów. Ze względu na ograniczoną objętość tego opracowania, bardziej szczegółowo przedstawiono tutaj jedynie dane odnośnie do przemysłu informatycznego i podzespołów będących jego zapleczem.

Początek bieżącego roku zastał gospodarkę USA w stanie najdłuższej po wojnie recesji. Niektóre przedsiębiorstwa musiały ograniczyć zatrudnienie, inne zamroziły lub zmniejszyły zarobki pracowników. Starano się wyeliminować działy nie przynoszące zysku. Przedsiębiorstwa muszą ograniczać swe aspiracje — realizacja zamierzeń wymaga coraz większego wysiłku.

Aktualne prognozy przewidują wzrost dochodu narodowego w 1983 roku o 2,3% (po spadku w 1982 o 1,7%). Rynek urządzeń elektronicznych wykazał w minionym roku wzrost obrotów o 14% — do ponad 138 mld $. Na rok bieżący wzrost ten szacuje się na 15%, przy czym — ze względu na zmianę asortymentu, wartości dolara, inflację itp. — nie są to wielkości w pełni porównywalne. Obroty na rynku półprzewodników wzrosły w 1982 roku o 11,8%, a na rok bieżący przewiduje się wzrost o 22% — do sumy 9,6 mld $. Sprzedaż pozostałych elementów i podzespołów (rezystory, kondensatory, przekaźniki lampy itp.) rozwija się wolniej — wzrost o 6,7% w 1982 roku i o 11,5% (do 14,3 mld $) w 1983. Najszybciej rozwijał się przemysł informatyczny, osiągając w ubiegłym roku 17,7% wzrostu; w roku bieżącym przewiduje się 18% — do 61,7 mld $. Natomiast obroty rynku elektroniki „konsumpcyjnej" (głównie sprzęt radiotelewizyjny) wzrosły w roku ubiegłym tylko o 6,7%, a na 1983 przewiduje się 6% — do prawie 23 mld $.

W Europie Zachodniej obroty na rynku sprzętu elektronicznego wzrosły w 1982 roku o 11,5%, ale na rok bieżący przewiduje się wzrost jedynie o 7% — do prawie 53 mld $. W dziedzinie informatyki i sprzętu biurowego wzrost wyniósł w roku ubiegłym 12%, a w 1983 szacowany jest na 14,5% — do 20,6 mld $. Natomiast w elektronice konsumpcyjnej, po stosunkowo wysokim wzroście (o 15%) w 1982, w bieżącym roku — m.in. na skutek konkurencji krajów Dalekiego Wschodu — przewiduje się spadek o 4% — do 15,2 mld $.

Do Japonii recesja dotarła dopiero w zeszłym roku, kiedy to wzrost dochodu narodowego miał wynieść według przewidywań rządowych 3,4% (poprzednio przewidywano 5,2%), a według niezależnych ekonomistów 2,1%. Przewidywana poprawa stanu gospodarki USA w drugiej połowie tego roku powinna pomóc w utrzymaniu tego wskaźnika w 1983 na poziomie 3%. Obroty przemysłu elektronicznego wzrosły w 1982 o 9%, a w bieżącym roku przewiduje się wzrost o 12% — do 50,5 mld $, przy czym wskaźniki te dla sprzętu elektronicznego wynoszą 8,5 i 11% (do 37 mld $), a dla elementów i podzespołów 11 i 14% (do 13,6 mld $).

W tabeli 1 zestawiono wyniki i przewidywania rynku półprzewodników USA, Europy Zachodniej i Japonii.

USA

W 1982 wystąpił w USA spadek sprzedaży półprzewodników dyskretnych o 1,2%, lecz na rok bieżący przyjmuje się już wzrost o 6%. Dla układów scalonych wzrost ten wynosi odpowiednio 15,5 i 26%. Należy przy tym uwzględnić, że ceny wielu podzespołów, zwłaszcza układów pamięciowych o dużej pojemności, zostały znacznie obniżone, oraz że ich wzrost ilościowy jest coraz większy. W drugiej połowie 1982 cena dynamicznych pamięci o swobodnym dostępie i pojemności 64 K wahała się w granicach

Tabela 1. Wyniki i przewidywania obrotów rynku półprzewodników (w mln $)

	USA				Europa Zachodnia			Japonia		
	1981	1982	1983	1986	1981	1982	1983	1982	1982	1983
Łącznie półprzewodniki	7007	7860	9614	16 551	2292	2436	2758	3534	3832	4572
Elementy dyskretne	1338	1322	1400	1 719	607	771	831	1179	1160	1251
diody	487	486	510	624	316	313	327	370	361	382
tranzystory	712	708	753	921	332	350	373	706	664	725
Układy scalone	5416	6277	7923	14 432	1400	1519	1761	2110	2372	2995
standardowe układy logiczne	1114	1183	1315	1 916				350	395	549
mikroprocesory i mikrokomputery	808	1053	1425	2 823	154	174	221	269	370	473
pamięci	1771	2113	2926	5 449	267	308	375	370	447	561
o dostępie swobodnym	873	1080	1656	2 748	162	193	231	182	220	271
dynamiczne	467	589	1065	1 468				106	127	160
statyczne	402	488	586	1 270				75	93	111
stałe	838	950	1139	2 280	105	116	144	139	179	226
liniowe	790	868	989	1 704	412	427	471	661	696	848
Układy na zamówienie	337	426	584	1 411						
Układy do urządzeń rynkowych (kalkulatory, zegarki, zabawki)	442	497	515	818	40	44	52	431	426	513
Urządzenia optoelektroniczne	253	261	291	400	91	103	114	246	277	326

4—6,5 $ i — mimo spadku z prawie 10 $ w 1981 — obroty wzrosły w 1982 o 153%. Na rok bieżący przewiduje się wzrost o ponad 200%, mimo przewidywanego spadku ceny poniżej 3$. Również znacznie zmalała cena pamięci statycznych o pojemności 16 K i wynosiła 5—8 $ w pierwszej połowie ubiegłego roku, a pod koniec 1982, przy dużych zamówieniach, wytwórcy japońscy sprzedawali je nawet po 3,25 $. Sprzedaż dynamicznych pamięci o swobodnym dostępie oraz szybkich pamięci statycznych wzrosła w roku ubiegłym, odpowiednio o 26 i 21%, a przewidywany wzrost w 1983 wynosi aż 81 i 20%.

Obroty standardowymi układami scalonymi wzrosły natomiast w 1982 r. tylko o 3%, natomiast na rok bieżący przewiduje się wzrost o 11%, głównie dzięki szybkim układom CMOS, których produkcja zwiększy się ponad trzykrotnie osiągając w 1983 r. 40 mln $ i 180 mln w 1986 r. W grupie pamięci stałych, pamięci programowane (wymazywalne) po gwałtownym spadku ich cen w 1981 — zanotowały w roku ubiegłym wzrost obrotów o 17,4%, a na 1983 r. przewiduje się wzrost o 25%. Jednocześnie tempo wzrostu pamięci trwałych wytwarzanych za pomocą masek (wykorzystywanych m.in. w zabawkach i grach) nieco osłabnie — z 22,6% w 1982 r. do 19% w roku bieżącym. Ciekawe, że pamięci na układach o sprzężeniu ładunkowym (CCD), którym niedawno przepowiadano bujny rozwój, są stopniowo eliminowane z rynku (z 12,5 mln $ w 1981 r. do 4,5 mln w 1986 r.), natomiast pamięci na cylindrycznych domenach magnetycznych zanotowały w roku ubiegłym wzrost o 54%, a na 1983 r. przewiduje się dalsze 70% — do 124,2 mln $. Zastępują one pamięci dyskowe w urządzeniach przenośnych i lotniczych.

Układy wytwarzane według życzeń odbiorcy (ang. custom IC) wykazały wzrost obrotów w 1982 r. o 48%, a na rok bieżący przewiduje się ich wzrost o 37,5%, przy czym w roku ubiegłym matryce programowane stanowiły ponad 30% produkcji, natomiast obecnie przewiduje się najszybszy rozwój sprzedaży matryc z elementów standardowych (z 21,1 mln $ w 1981 do 482 mln w roku 1986, kiedy stanowić będą one 34% układów tego typu). Wskaźniki wzrostu sprzedaży układów liniowych wynoszą odpowiednio (1982 i 1983) 10 i 14%, a dla urządzeń optocyfronicznych 3 i 12%.

W dziedzinie pozostałych elementów i podzespołów rozwój odbywa się w wolniejszym tempie. Wzrost w roku ubiegłym wyniósł tylko 6,7%, choć w 1983 przewiduje się wzrost o 12% — do 14,3 mld $. Obroty elementami biernymi na ogół spadły w 1982 (kondensatory o 8%, łączówki o 1,5%, tylko dla rezystorów zanotowano wzrost o 5%). W roku bieżącym przewiduje się tu poprawę (wzrost dla kondensatorów o 14% do 1,24 mld $, łączówek o 10,3% do 2,26 mld i rezystorów o 11% do 860 mln). Dla przekaźników wskaźnik wzrostu utrzymuje się na poziomie 4%, a dla przełączników i klawiatur ok. 8% — do 1,1 mld $ w 1983. Rynek lamp elektronowych, którego ponad połowę stanowią kineskopy telewizyjne, utrzymuje średnie tempo wzrostu 8% (do 1,7 mld $ w 1983), przy czym wzrost ten dla kineskopów wynosi 5—6%, dla lamp z falą bieżącą 28,5% w 1982 i 21% w 1983, natomiast dla lamp odbiorczych obserwujemy spadek o 11—12% rocznie. Po wysokim wzroście (o 39%) w dziedzinie podzespołów hybrydowych w roku ubiegłym, przewiduje się tu 15% wzrostu w 1983 oraz średni wzrost o 24,4% rocznie do 1986 (do 916 mln $).

Przemysł informatyczny interesujący nas najbardziej, charakteryzował się w 1982 roku nieco przyhamowanym rozwojem. Wyniki i przewidywania rozwoju poszczególnych dziedzin tego przemysłu przedstawiono w tabeli 2. Niektóre działy, jak choćby komputery osobiste, osiągnęły w USA wzrost o 100% w 1982, a na rok bieżący przewiduje się rozwój o 62,5% i 45,4% rocznie — do 20 mln $ w 1986. Podobnie jednostki małych pamięci dysko-

Tabela 2. Wyniki i przewidywania rozwoju obrotów przemysłu informatycznego (w mln $)

	USA				Europa Zachodnia			Japonia		
	1981	1982	1983	1986	1981	1982	1983	1981	1982	1983
Łącznie; systemy komputerowe, urządzenia peryferyjne i wyposażenie biurowe	44 312	54 145	61 712	105 140	16 093	18 025	20 646	10 078	12 130	14 007
Łącznie systemy sprzedawane i dzierżawione	16 625	20 058	24 227	45 900	10 967	12 543	14 500	3 517	4 222	4 854
małe	3 606	5 908	8 877	23 700	3 749	4 538	5 619	1 156	1 594	2 036
średnie	4 200	4 500	4 900	6 500	3 248	3 521	3 750	909	1 024	1 164
duże	8 090	8 500	8 900	12 200	3 971	4 484	5 131	1 452	1 604	1 654
Mini- i mikrokomputery	5 600	6 750	8 126	14 170				1 075	1 598	2 033
Pamięci operacyjne	817	853	923	1 309	82	91	102	244	277	313
Pamięci masowe	2 685	3 286	4 292	7 563				1 186	1 319	1 496
na dyskach stałych	550	780	1 315	2 860						
na dyskach elastycznych	430	600	840	1 850						
na taśmach magnetycznych	335	417	521	933						
Urządzenia wejścia-wyjścia	3 461	4 083	4 804	7 682	712	800	887	1 104	1 457	1 550
Terminale	3 807	4 552	5 686	11 414	1 533	1 742	2 028	958	992	1 080
Systemy zbierania danych źródłowych	1 722	1 846	2 002	2 750	226	252	304	504	566	640
Automatyzacja biur	9 355	10 492	11 452	14 202	2 417	2 428	2 606	1 330	1 521	1 830
Łącznie oprogramowanie	2 149	3 105	7 192	16 810						
użytkowe	508	751	1 018	3 393						
wspomagania projektowania	312	451	745	1 801						
testująco-diagnostyczne	329	493	644	1 412						
języki wysokiego poziomu	420	541	610	1 201						
systemy operacyjne	520	870	4 176	9 003						

wych (elastycznych i typu Winchester), gdzie wskaźnik wzrostu utrzymuje się na poziomie 40% do 840 mln $ w 1983. W dalszej kolejności należałoby wymienić systemy wspomagania i automatyzacji, gdzie tegoroczny wzrost oceniany jest na 35% — do 1,55 mld $. Oceniając rynek pamięci operacyjnych, należy uwzględnić, że jest on od kilku lat zdominowany przez pamięci półprzewodnikowe (77% w roku 1981 i 94% w 1986).

Przemysł ten charakteryzuje się w USA szczególnie ostrą walką konkurencyjną — małe komputery oferuje ponad 150 dostawców, a łączna liczba krajowych i zagranicznych firm produkujących sprzęt komputerowy i teletransmisyjny oraz świadczących usługi informatyczne przekracza 1500.

Coraz bardziej rozwija się rynek oprogramowania. Jego wzrost w ubiegłym roku wyniósł 45%, a do 1986 powiększy się ponad pięciokrotnie. W szacunku tym należy ponadto uwzględnić oprogramowanie nieujawnione (podstawowe, pomocnicze i serwisowe), którego wartość można było w 1982 roku ocenić na ponad 45 mld $. Do 1986, kiedy wartość oprogramowania związanego z systemem osiągnie połowę wartości sprzętu — będzie to kwota przekraczająca 100 mld $. Największą dynamikę wykazują tu systemy operacyjne — wzrost w 1982 o 67% i wyjątkowo wysoki, prawie pięciokrotny wzrost w roku bieżącym, na skutek zarówno wzrostu wymagań użytkowników jak i zwiększenia dostępności sprzętu w postaci tanich mikrosystemów. W następnych latach wzrost ten szacowany jest na 30% rocznie. Wskaźniki dla języków wysokiego poziomu wynoszą 30% w roku 1982 i 13% w bieżącym. Zmniejszenie tempa rozwoju wynika tu z wzrastającej dostępności gotowych pakietów programowych. Oprogramowanie użytkowe wzrosło o 48% w roku ubiegłym, w 1983 przewiduje się 36% wzrostu oraz prawie po 50% rocznie w latach następnych. Oprogramowanie systemów wspomagania rozwijało się w tempie 44% w 1982 i 65% w roku obecnym, natomiast do celów testowania i diagnostyki — odpowiednio 50 i 31%. W następnych latach mają one rozwijać się w tempie ponad 30% rocznie.

Obroty rynku elektrotechnicznych przyrządów pomiarowych w USA wykazały 14% wzrostu w 1982, a na rok bieżący przewidywane jest 17% — do 4,2 mld $, przy czym najbardziej rozwijają się automatyczne systemy testujące, które po umiarkowanym wzroście o 14,4% w 1982, w 1983 mają osiągnąć dwukrotnie wyższe tempo — do 1,42 mld $. Wśród nich najwyższe tempo rozwoju utrzymują międzyoperacyjne testery układów scalonych, których sprzedaż wzrosła w 1982 o 28%, a w roku bieżącym przewiduje się wzrost o 52% — do 58,8 mln $. Również pomyślnie rozwijała się produkcja testerów pakietów, zwłaszcza diagnostycznych: wzrost o 27% w roku ubiegłym i przewidywane 37,2% w 1983 — do 189 mln $. Rozwój testerów funkcjonalnych odbywa się wolniej, natomiast testery kombinowane rozwijać się będą jeszcze szybciej i w roku 1986 będzie się ich sprzedawać więcej niż diagnostycznych.

W grupie przyrządów pomiarowych największy rozwój wykazują systemy wspomagania projektowania, analizatory logiczne i emulatory, które pochłaniają 20% inwestycji w tej dziedzinie. Systemy wspomagania wykazały w ubiegłym roku wzrost o 32,5%, przy czym szybciej rozwijają się systemy uniwersalne, co obecnie ma się jeszcze bardziej uwidocznić. Wśród oscyloskopów zaś — począwszy od bieżącego roku — szybciej rozwijają się szerokopasmowe (ponad 100 MHz).

W zakresie elektroniki konsumpcyjnej rynek odbiorników telewizyjnych niemal się nie zmienił w 1982 r., a na rok bieżący przewiduje się niewielki wzrost (3,5%), przy czym produkcja odbiorników czarno-białych maleje. Podobnie przedstawia się sytuacja ze sprzętem elektroakustycznym, zwłaszcza wysokiej jakości, gdzie w roku ubiegłym odnotowano nawet pewien spadek. Jedynie magnetowidy notują bujniejszy rozwój — wzrost o 29% w 1982 i przewidywane 17,5% na rok bieżący — do 1,5 mld $, a już zupełnym wyjątkiem są przenośne magnetofony kasetowe, obroty którymi podwoiły się w roku ubiegłym do 476 mld $; niemniej rynek już się nasycił i od 1983 przewidziany jest powolny spadek.

Sprzęt telekomunikacyjny rozwija się natomiast w dobrym tempie. Rok 1982 przyniósł wzrost o 11%, a na rok bieżący przewidywany jest wzrost o 17% — do 8,3 mld $. Wiąże się to z przyjęciem standardów dla sieci lokalnych, wykorzystaniem magnetowidów i komputerów osobistych, a także opracowywaniem dużych systemów światłowodowych i rozwojem komunikacji satelitarnej. Przykładowo — rynek modemów wzrósł o 18% w roku ubiegłym i przewidywany na 1983 wzrost wynosi 35% — do 1,1 mld $, natomiast systemy światłowodowe wykazują w bieżącym roku 65% wzrostu (do 570 mln $).

Elektroniczna aparatura medyczna i analityczna rozwija się nierównomiernie. Tempo rozwoju części analitycznej wynosi ponad 11% i nieco słabnie (w bieżącym roku do wartości 1,3 mld $), przy czym wyróżnić można chromatografy, zwłaszcza cieczowe, których przyrost powiększa się dwukrotnie do 28% i wartości 256 mln $ w 1983. Natomiast aparatura medyczna wykazuje wzrost o 15,2% w 1982 i ponad 19% — do 4,1 mld $ w roku bieżącym. Najbardziej dynamicznie rozwija się dział diagnostyczno-terapeutyczny: tomografy (16,3% w 1982 i 20% do 955 mln $ w 1983), skanery ultradźwiękowe (odpowiednio 17,6 i 30%) oraz automatyczne analizatory krwi (17 i 20%).

Obroty rynku elektroniki przemysłowej wykazują zmniejszenie tempa wzrostu z 17,5% w roku ubiegłym do 16,5% w 1983, osiągając 4,5 mld $. Dzieje się tak na skutek ograniczeń inwestycji. Jednakże systemy robotów przemysłowych wykazały wzrost o 13,5% w 1982, a w roku bieżącym oczekuje się wzrostu o 34% — do 290 mln $. Zmniejsza się natomiast rozwój systemów sterowania procesami (z 15% w 1982 do 8% i 1,6 mld $ w 1983), podczas gdy systemy numerycznego sterowania wzrastają odpowiednio o 33,5 i 40% — do 372 mln $ w 1983.

W przypadku linii technologicznych do produkcji półprzewodników w ubiegłym roku zanotowano spadek (3%) w związku z przedłużającym się oczekiwaniem na poprawę koniunktury, przy czym nie dotyczyło to wytwarzania masek (wzrost o 20%). Rok 1983 ma przynieść 13% wzrost — do 991 mln $.

Wydatki rządowe na elektronikę wzrosły w 1982 o 17% i wzrastają w roku bieżącym o 11% — do 34,6 mld $, z czego 90,6% stanowią wydatki wojskowe. W ramach tych wydatków fundusze na badania wzrastają wolniej niż poprzednio (9% — do 9,2 mld $ w roku 1983), natomiast na elektronikę samolotową szybciej (o 23% — do 5,1 mld $). Jeszcze większy wzrost przewidywany jest w latach następnych na elektronikę w wojskowych systemach kosmicznych. Inne pozycje wydatków rządowych, jak elektronika dla służby zdrowia i oświaty, ograniczone są w bieżącym roku do poziomu o 1% niższego niż w 1982.

EUROPA ZACHODNIA

Europa Zachodnia zwiększyła w 1982 swe obroty w dziedzinie półprzewodników o 6%, a na rok bieżący przewidziany jest wzrost o 13% — do wartości 28 mld $. Całkowite obroty rynku elementów i podzespołów wzrosły w ubiegłym roku o 5%, natomiast w 1983 r. przewidziany jest wzrost o 8% — do 10,2 mld $, z czego 31% wypada na RFN, 26% na Wielką Brytanię, a 20% na Francję. Procentowy wzrost jest mniejszy niż kilka lat temu. W RFN w drugiej połowie 1982 sprzedaż półprzewodników zmniejszyła się. We Włoszech przygotowywane jest wprowadzenie na rynek ok. 100 układów CMOS opracowywanych wspólnie z japońską firmą TOSHIBA. Hiszpania opanowała już technologię produkcji układów LSI. Rynek obwodów scalonych zwiększył obroty o 8% w 1982 i w roku bieżącym przewiduje wzrost o 16% — do 1,8 mld $, przy czym układy liniowe (odpowiednio 3,5 i 10%) i standardowe cyfrowe (7 i 14%) rozwijają się wolniej, natomiast szybciej — pamięci (15 i 22% — do 375 mln $) oraz mikroprocesory i mikrokomputery (13 i 27% — do 221

mln). Wreszcie urządzenia optoelektroniczne nieco zwalniają swój rozwój — z 12,8% w roku ubiegłym do 10,6% (114 mln $) w 1983, a lampy trochę przyspieszają swe powolne tempo z 3% w 1982 do 5% (1,6 mld $) w roku bieżącym.

Rynek informatyczny Europy Zachodniej jeszcze kilka lat temu był bardzo dynamiczny. Obecnie spadek realnej wartości dolara i osłabienie tempa rozwoju gospodarki czołowych państw zachodnioeuropejskich, zmieniło ten stan. Wolniej rozwijają się zwłaszcza systemy komputerowe średniej wielkości, których przyrost w tym roku ocenia się w RFN na 4—5%. Komputery osobiste, tak jak wszędzie, rozwijają się natomiast błyskawicznie (przewidywany wzrost o 79% w roku bieżącym, o 43% w roku przyszłym i o 41% — do 290 tys. sztuk — w 1985). W Wielkiej Brytanii firma SINCLAIR wytwarza miesięcznie 40 tys. mikrokomputerów ZX i skutecznie konkuruje w tej dziedzinie z innymi przedsiębiorstwami. Natomiast potężna dotąd firma ICL musiała już przed rokiem zwrócić się do rządu o pomoc. We Francji następuje koncentracja przemysłu informatycznego w ramach firmy CII-HONEYWELL BULL, która przejęła od koncernu THOMSON-CSF specjalizującą się w mini- i mikrosystemach firmę SEMS. We Włoszech, mimo słabej kontroli i koordynacji ze strony rządu, przewiduje się rozwój o średnio 25%, a w komputerach osobistych — wzrost o 75% w tym roku.

Rynek elektroniki konsumpcyjnej w Europie Zachodniej wzrósł natomiast w 1982 o 9,5% do 15,9 mld $, ale na rok bieżący przewiduje się czteroprocentowy spadek, przy czym np. rynek magnetowidów wykazał 44% wzrost w roku ubiegłym, a przewidziany przyrost na 1983 wynosi 6% — do 2,6 mld $ (w RFN i Wielkiej Brytanii co dziesiąte gospodarstwo domowe posiadało magnetowid w końcu 1982). Dużą część tej dziedziny elektroniki stanowią wciąż kolorowe telewizory, których wyprodukowano w zeszłym roku prawie 11 mln szt. (wzrost o 1,7%) i przewiduje się utrzymanie podobnego tempa również w 1983 roku.

W czterech największych krajach Europy Zachodniej 60% rodzin posiadało już telewizor kolorowy, a w 1986 wskaźnik ten ma wzrosnąć do 80%. Natomiast na rynku urządzeń elektroakustycznych przewiduje się w tym roku spadek obrotów o 9,7%.

Telekomunikacja w Europie Zachodniej rozwija się w tempie od 8% (1982) do 10% (przewidywania na 1983 — do ponad 11 mld $. Największym klientem jest tu poczta, dużym — telewizja przewodowa. We Francji rozwija się bujnie komunikacja światłowodowa — do roku 1986 sieć światłowodowa ma objąć milion gospodarstw domowych. Podobne działania podejmuje się w Wielkiej Brytanii.

Zachodnioeuropejski rynek elektronicznych przyrządów pomiarowych wykazuje podwojenie tempa wzrostu — do 13% i 1,2 mld $ w roku bieżącym. Podobnie jak w USA, mikroprocesorowe systemy wspomagania rozwijają się najszybciej (ok. 15% — do ponad 150 mln $ w 1983).

JAPONIA

Japonia, jak wspomniano, odczuła recesję w połowie zeszłego roku w postaci spadku eksportu o ponad 10%, zwłaszcza do krajów OPEC i Ameryki Łacińskiej. Obiekty przemysłu hutniczego i petrochemicznego były wykorzystane zaledwie w 50—60%. Zmalały zamówienia na obrabiarki i statki, a liczba sprzedawanych samochodów, mimo niskich cen, spadła. Obroty w domach towarowych spadły o 2%, a bezrobocie osiągnęło 2,3%. Obroty rynku elementów i podzespołów elektronicznych wykazały w 1982 r. wzrost o 11%, a na rok bieżący przewiduje się o 14% do 13,6 mld $, przy czym podzespoły bierne i mechaniczne zanotowały w ubiegłym roku wzrost o 11,8%, a na 1983 spodziewany jest wzrost o 10,5% — do 7,3 mld $. Rynek półprzewodników przedstawiony jest w tabeli 1.

Dynamiczne pamięci o swobodnym dostępie i pojemności 256 K jeszcze nie pojawiły się masowo: dopiero w końcu bieżącego roku realizowane będą zamówienia po kilkadziesiąt tysięcy sztuk, a konkurencji cen należy się spodziewać w 1984 r. Na razie tanie kostki o mniejszej pojemności (3 $ za 64 K i powyżej dolara za 16 K) powodują ogromne zamówienia i nieprzerwaną pracę fabryk z ograniczeniem jednak produkcji pamięci 16 K. Pamięci statyczne rozwijały się jeszcze szybciej (23,5% wzrostu w roku ubiegłym, przy 19,5% dla pamięci dynamicznych), lecz zostały nieco przyhamowane przez producentów, którzy chcieli uniknąć zbyt gwałtownego spadku cen (pamięci statyczne są już tylko 20% droższe od dynamicznych).

Liczby charakteryzujące poszczególne działy japońskiego przemysłu informatycznego podane są w tabeli 2. Ogólnie można stwierdzić zmniejszanie się tempa wzrostu kosztu sprzętu oraz tendencję do przechodzenia na systemy przetwarzania rozproszonego, co prowadzi do ściślejszych związków z przemysłem telekomunikacyjnym. Podobnie wzrastają powiązania pomiędzy elektroniką konsumpcyjną a automatyzacją biur i produkcją terminali. Podstawowymi wymaganiami jest łatwość obsługi sprzętu oraz dostosowanie systemów przetwarzania tekstów do innych alfabetów, w tym również chińskiego. Wskaźnik wzrostu sprzedaży komputerów biurowych przekracza 25%, natomiast dużych systemów poniżej 10%.

Dotychczasowy, szczególnie szybki rozwój elektroniki konsumpcyjnej został zahamowany. W 1982 nastąpił spadek obrotów o 2%, a 1% wzrostu w tym roku nie przywróci jeszcze poziomu 11,2 mld $ w 1981. Sprzęt elektroakustyczny wzrasta obecnie o 3% — do 3,7 mld $, ale kosztem urządzeń wyższej jakości (stereo), natomiast magnetofony rozwijają się szybciej. Kolorowych telewizorów sprzedano w roku ubiegłym 6,9 mln, w większości małych. Rośnie produkcja magnetowidów (na rynek wewnętrzny z 1,65 mln szt. w roku 1981 do 2,7 mln w tym roku), lecz nie jest to rynek o dużej rentowności.

Osłabione tempo innych gałęzi wpłynęło też na rynek przyrządów pomiarowych, który wzrósł o 10% w 1982, a przewidywany przyrost na rok bieżący wynosi tylko 11% — do 838 mln $. Próbuje się ożywić go przez rozwój inteligentnych pisaków x-y dostosowanych do mikro- i minikomputerów oraz systemów pomiarowych. W zeszłym roku sprzedaż analizatorów Fouriera wzrosła o 30%, a na 1983 r. przewidziany jest wzrost o 17%. Testery obwodów scalonych i pakietów wytwarzane są w Japonii przez filie firm amerykańskich, jak GENRAD, HEWLETT-PACKARD i FAIRCHILD. Wzrasta również produkcja oscyloskopów, zwłaszcza szerokopasmowych.

Rynek telekomunikacyjny po niewielkim (niespełna 1%) spadku obrotów w roku ubiegłym ma wzrosnąć w 1983 o 18% — do prawie 3 mld $. Rozpoczyna działanie wielokanałowy system połączeń handlowych, pozwalający na kontakt dwóch stacji za pomocą stałej stacji bazowej. Po uruchomieniu tego systemu przewidziany jest wzrost w ciągu najbliższego dziesięciolecia o 30% rocznie. Natomiast od ubiegłego roku nastąpiło ograniczenie rozwoju centrali telefonicznych i sprzętu transmisyjnego. Dopiero przy przejściu w 1985 na centrale o koncepcji cyfrowej przewiduje się ponowny wzrost tej dziedziny. Automatyzacja biur wzbudza zainteresowanie sieciami lokalnymi, lecz na razie stosowane są one tylko w ramach przedsiębiorstw na łączach światłowodowych.

Ostatni z omawianych działów — elektronika przemysłowa wykazał w 1982 r. wzrost obrotów o 10%, natomiast na rok bieżący przewidziany jest wzrost o 17% do 4,1 mld $, przy czym zamówienia na urządzenia do produkcji półprzewodników w roku ubiegłym wzrosły aż o 57%, a wzrost w 1983 r. przewiduje się na poziomie 49% — do 670 mln $. Dla automatyki wskaźniki te wynoszą odpowiednio 24 i 26% do 653 mln $, przy czym najszybciej rozwijają się roboty przemysłowe, natomiast wolniej — sterowanie procesami, które stanowi już tylko 25% tej dziedziny elektroniki (w 1981 — 30%).

Opracował JAN RYŻKO
na podstawie
ELECTRONICS nr 1/1983

Istniejący już od ponad 50 lat amerykański dwutygodnik ELECTRONICS poświęcił — w październiku ub.r. — jeden numer podsumowaniu stanu techniki w wielu dziedzinach — między innymi w tych, które bezpośrednio dotyczą informatyki. Chcąc informować Czytelników o tym, co dzieje się obecnie w informatyce światowej, postanowiliśmy przedstawić na naszych łamach niektóre zamieszczone tam artykuły, opisujące główne kierunki postępu w dziedzinie sprzętu i oprogramowania. Na początek przedstawiamy stan technologii półprzewodnikowej i konstrukcji układów scalonych. W następnych numerach — o oprogramowaniu mikrokomputerów i urządzeniach peryferyjnych. (Red.).

Mikrosystemy

Wśród zagadnień występujących przy projektowaniu systemów, tak sprzętu, jak i oprogramowania, dominują problemy związane z projektowaniem architektury oraz z wydajnością i elastycznością. Pojawiły się układy o architekturze mikroprogramowej, próbujące wypełnić luki między procesorami 8-, 16- i 32-bitowymi i wkroczyć w rejon dostępny dotąd tylko dla oprogramowania użytkowego. Wymagania stawiane potrzebnym w małych ilościach układom zamawianym (ang. custom) i specjalizowanym próbowano rozwiązać dwojako: przez użycie mikrokodowania lub przez zestawianie tych układów z wcześniej zaprojektowanych elementów standardowych.

Wśród układów specjalizowanych, wprowadzonych na rynek w ubiegłym roku, główną pozycję zajmują układy sterowania monitorami znakowymi i graficznymi oraz procesory arytmetyczne i przeznaczone do przetwarzania bardzo szybkich sygnałów cyfrowych. Rozpowszechnienie układów specjalizowanych pogłębia tendencję do przetwarzania zdecentralizowanego (ang. distributed processing), ale w pełni rozwinięta architektura zdecentralizowana jest projektowana o wiele wolniej.

Rosnące możliwości procesorów 16-bitowych, w porównaniu z popularnymi 8-bitowymi, spowodowały szybkie ich zaakceptowanie. O wyborze określonego procesora 16-bitowego będzie decydować dostarczane do niego wyposażenie dodatkowe.

W ubiegłym roku na czoło wysunęły się układy INTELA, czego przykładem jest wybór przez IBM procesora 8088 do nowego wyrobu, jakim jest PERSONAL COMPUTER. Można się jednak spodziewać rywalizacji z procesorami 68000 MOTOROLI i Z8000 ZILOGA. Główni producenci, zmierzając do zwiększenia produkcji, usiłują uporządkować zarówno sprawy oprogramowania, jak i zapewnić sobie dostawy układów pomocniczych z innych źródeł.

Dążąc do zwiększania wydajności w oparciu o istniejące oprogramowanie, firma INTEL wprowadziła na rynek 16-bitowy procesor typu 80286. Procesor ma architekturę potokową i niespotykane dotąd rozwiązanie, polegające na umieszczeniu w układzie jednostki zarządzania pamięcią (ang. memory management). Zapewnia to sześciokrotny wzrost jego wydajności w porównaniu z procesorem typu 8086. Układ 80286 zachowuje zgodność kodu wewnętrznego ze swym poprzednikiem i umożliwia przeniesienia oprogramowania użytkowego.

Cechą, która może okazać się decydująca przy wyborze procesora 16-bitowego, jest wprowadzona w układzie typu 80286 metoda zarządzania pamięcią. Firma INTEL stosuje wirtualne adresowanie pamięci za pomocą segmentacji sprzętowej. Oznacza to, że programy są zawarte w logicznie oddzielonych sekcjach lub segmentach. W przeciwieństwie jednak do klasycznych komputerów, gdzie segmenty mogą być tak duże, jak efektywny rozmiar pamięci, segmenty INTELA mają pojemność 64 K bajtów. Chociaż w większości przypadków zawężenie to nie może być czynnikiem ograniczającym, na pewno będzie wpływać na elastyczność w takich zastosowaniach, jak programy zarządzania bazami danych.

Interesującym procesorem 16-bitowym (o wewnętrznej organizacji 32-bitowej, lecz na zewnątrz 16-bitowy) jest wprowadzony na rynek w ubiegłym roku NS16032 produkcji NATIONAL SEMICONDUCTOR, w którym zastosowano żądanie stronicowania (ang. demand paging) do adresowania wirtualnego. Obszar adresowy tego procesora jest podzielony na strony o długości 512 bajtów. Gdy jednostka zarządzania pamięcią wykryje adres strony nieobecnej, wykonuje procedurę wymiany, sprowadzając potrzebną stronę z pamięci masowej w miejsce strony niewykorzystywanej.

W obu przypadkach, gdy jest adresowana nieobecna strona lub segment, procesor musi odłożyć (lub zawiesić) wykonanie instrukcji, która spowodowała odwołanie do ostatnio pobranego kodu lub danej. Później procesor ponownie wykonuje tę instrukcję (lub kontynuuje jej wykonanie). W procesorach, które mają odrębne jednostki zarządzania pamięcią znajdujące się w innym układzie, tak jak NS16032, wyprowadzany jest z układu sygnał, który umożliwia jednostce centralnej odrzucenie instrukcji odwołującej się poza dostępną pamięć. Firma ZILOG (wchodząca w skład wielkiego koncernu EXXON) dołączyła sygnał odrzucenia instrukcji do układu Z8003; także procesor 68000 MOTOROLI ma taki mechanizm.

Oprócz procesora NS16032 z wewnętrzną architekturą 32-bitową, któremu dorównuje jedynie procesor 68000, firma NATIONAL SEMICONDUCTOR zapowiedziała wprowadzenie pełnej gamy układów i programów dodatkowych. Oferowane będą układy do zarządzania pamięcią, obliczeń zmiennoprzecinkowych, sterowania przerwaniami oraz sterowania bezpośrednim dostępem do pamięci (DMA), przy czym będą one w przyszłości dostarczane także do 8- i 32-bitowej odmiany procesora. Oprogramowanie podstawowe obejmuje translatory języków PASCAL, FORTRAN i PL/I (TRANSLATIONS SYSTEM) oraz C (BELL LABORATORIES), a także systemy operacyjne VRTX (HUNTER READY), CP/M (DIGITAL RESEARCH) i UNIX (BELL LABORATORIES).

Ostatnio firma INTEL wprowadziła na rynek procesor zmiennoprzecinko-

wy typu 80287 i układ z systemem operacyjnym CP/M — oba przeznaczone do współpracy z procesorem typu 80286. MOTOROLA również powiększy bazę układów pomocniczych dzięki zawartym w ubiegłym roku (z firmami MOSTEK, SIGNETICS, ROCKWELL, PHILLIPS i HITACHI) porozumieniom o produkcji układów dostosowanych do współpracy z procesorem typu 68000.

Firma NCR rozpoczęła produkcję procesora NCR/32. Jest to zestaw czterech układów łączących cechy serii BALLMAC-32 firmy BELL LABORATORIES i serii iAPX 432 firmy INTEL. Procesor ten jest jednym z najbardziej obiecujących osiągnięć w dziedzinie mikrosystemów - wprowadza elastyczność architektury. Dzięki redukcji liczby cykli potrzebnych do wykonania instrukcji, NCR/32 będzie mógł np. wykonać 32-bitowe dodawanie międzyrejestrowe w pojedynczym cyklu zegarowym.

Zestaw NCR/32 jest zewnętrznie mikroprogramowany. Ta właściwość wraz ze 128 K-bajtową pamięcią mikrorozkazów i czasem cyklu procesora wynoszącym 150 ns umożliwia bardzo wydajną emulację różnych zbiorów instrukcji. Projektanci mogą więc użyć tego zestawu, po wpisaniu do układu pamięci odpowiedniego mikrokodu, do wykonywania kodu wynikowego zaczerpniętego z istniejącego oprogramowania użytkowego. NCR ocenia, że — na przykład — maszyna wirtualna COBOLU do wykonywania programów napisanych w tym języku mogłaby być zaimplementowana już w 25 K bajtach mikrokodu.

Niektóre systemy stosują prostsze, lecz bardziej ograniczone, środki wykorzystania istniejącej bazy oprogramowania 8-bitowego za pomocą zwykłego włączenia procesora 8-bitowego do systemu — razem ze stosowanym procesorem 16-bitowym. Przykładem jest pakiet Baby Blue CPU Plus z jednostką centralną Z80 CPU, która zapewnia, że PERSONAL COMPUTER firmy IBM uzyskuje dostęp do olbrzymiego oprogramowania tego mikroprocesora.

Architektura procesora NCR/32 jest ogólniejszym rozwiązaniem problemu zgodności oprogramowania. Zastosowanie tego kompletu układów jest jednak uzależnione od możliwości dostarczenia lub przebudowywania przez firmę NCR mikrokodu, lub udostępnienia takiego systemu projektowania, który pozwoliłby programistom projektować własny mikrokod.

Jednym z ciekawszych rozwiązań jest architektura opracowana w laboratoriach badawczych firmy TEXAS INSTRUMENTS — może być ona rekonfigurowana dynamicznie.

Restrukturalizowane projektowanie układów scalonych jest oparte na procesorze 16-bitowym (nazywanym segmentem mikroprogramowanym), jednostce sterowania przepływem danych przez zestawiany zbiór ścieżek oraz jednostce przydzielania pamięci, która zarządza zgłoszeniami z dowolnego z procesorów dołączonych do magistrali. Firma TEXAS INSTRUMENTS (TI) ma zamiar umieścić cztery mikroprogramowane segmenty w jednym układzie, gdzie mają być one połączone — tak, by mogły pracować w wielu trybach. Na przykład — segmenty mogą być połączone w pary i tworzyć dwa procesory 32-bitowe do przetwarzania równoczesnego.

TI rozpatruje możliwość wykonania wersji mikroprogramowanej komputera VAX 11/780 firmy DEC. Na podstawie symulacji opartej na układach prototypowych, TI szacuje, że pojedyncze restrukturalizowane układy scalone, zestawione w podany sposób, będą mogły wykonywać instrukcje komputera VAX 11/780 od dwóch do czterech razy szybciej. Tak jak w przypadku zestawu układów NCR, mikroprogramowanie takiego sprzętu może okazać się przeszkodą w jego zaakceptowaniu aż do czasu, gdy dostępne będą odpowiednie pomoce programowe.

Wprowadzany z opóźnieniem układ TMS 99000 umożliwia mikroprogramowanie, choć w bardziej ograniczonej formie. Mikrokod przechowywany w pamięci stałej (ROM) umieszczonej w układzie, nazywanej przez TI MACROSTORE, uzupełnia standardowy zbiór instrukcji instrukcjami specjalizowanymi o dużej wydajności. Pamięć MACROSTORE jest programowana przez TI kodem klienta w czasie produkcji (na poziomie projektowania maski). Możliwość zaprogramowania jest więc dostępna dla wytwórców OEM dążących do powiększania wydajności sprzętu specjalizowanego. Na przykład, wytwórcy OEM mogą zmieniać szybkość działania układu w takich zastosowaniach, jak grafika komputerowa, definiując specjalne do tego celu instrukcje podstawowe w pamięci MACROSTORE. Pamięć MACROSTORE, jako pamięć ROM, nie jest dostępna z zewnątrz. W ten sposób umieszczony w niej mikrokod, będący własnością klienta, jest skutecznie chroniony.

Inna technika, używana do kontrolowania dostępu z zewnątrz do wymazywalnej pamięci stałej wykorzystuje bit ochrony pamięci (ang. memory-protection bit). W układzie MC 68705P5 firmy MOTOROLA, po wpisaniu kodu do pamięci EPROM, zaprogramowanie bitu ochrony pamięci odcina zewnętrzny wgląd w jej obszar. Przy próbie zewnętrznego odczytu z pamięci EPROM wszędzie jest przywracany stan jedynki logicznej, co jest równoważne wykasowaniu zawartości pamięci.

Rosnąca liczba wytwórców sprzętu OEM powoduje wzrost zamówień na układy bardzo dużego stopnia scalenia (VLSI). Jest to sposób ochrony przez firmy własnego oprogramowania układowego. W rezultacie zwiększa się atrakcyjność systemów zbudowanych bez standardowych elementów. Przy zastosowaniu tej techniki jeden ze standardowych mikroprocesorów służy jako jądro, wokół którego dobudowywany jest określony zestaw komórek funkcjonalnych.

Większość procesorów specjalizowanych jest wykorzystywana w ściśle określonych zastosowaniach jako układy końcowe niskiego poziomu o intensywnym wykorzystaniu we-wy. Wykorzystanie idei komórki standardowej pozwala na skonstruowanie pojedynczego układu, który spełnia optymalnie wymagania we-wy. Obecnie katalogi zawierają bardzo dużo specjalizowanych mikrokomputerów jednoukładowych do ukierunkowanych (ang. dedicated) zastosowań.

Konstruktorzy wykorzystujący mikroprocesory w celu uzyskania lepszych możliwości kształtowania architektury systemu i jego elastyczności uważają, że układy specjalizowane tego typu stanowią istotny wkład w rozwój przetwarzania zdecentralizowanego. Oprócz układów przetwarzania numerycznego, projektowanych do współpracy z określonymi mikroprocesorami, korzystne będzie stosowanie do tego celu wprowadzonych w ostatnich latach układów przeznaczonych do pracy z różnymi jednostkami centralnymi.

Firma HEWLETT-PACKARD obrała kierunek na odciążenie jednostki centralnej przez przekazanie mikrosystemowi operacji zmiennoprzecinkowych. Grupa układów produkowanych przez tę firmę dokonuje podziału podstawowych operacji arytmetycznych na oddzielne części — jeden układ wykonuje dodawanie i odejmowanie, inny realizuje mnożenie, inny wreszcie — dzielenie.

Firma TEXAS INSTRUMENTS zapowiedziała wprowadzenie układu zdolnego do bardzo wydajnego przetwarzania sygnałów cyfrowych. Układ TMS 320 o architekturze typu HARVARD (konstrukcja z podwójną szyną do równoległego pobierania kodu i danej) wspomaga szybką arytmetykę zespoloną. Mnożenie zespolone może być wykonane w ciągu jednego cyklu zegara o okresie 200 ns. Do układu TMS 320 dostarczany będzie przez producenta także system projektowania.

W ostatnim okresie wzrasta zainteresowanie wysokowydajnymi systemami przetwarzania sygnałów o wysoko zrównoleglonej architekturze nazwanej „systolic data processing". Podobieństwo tej techniki, polegającej na przepompowywaniu danych przez sieć procesorów, do sposobu, w jaki serce przetacza krew przez naczynia krwionośne, użyczyło nazwy tej metodzie. Dobierając oddzielne procesory do pojedynczych kroków obliczeniowych w algorytmach wykorzystujących wektory lub tablice danych, „systolic processors" wykazują regularnie powtarzającą się strukturę, korzystną do

realizacji w technologii VLSI, i dużą przepustowość wymaganą do przetwarzania sygnałów. Badanie możliwości tej metody koncentruje się głównie na ustalaniu powiązań między sygnałami sterującymi przetwarzaniem.

Najciekawsze wyniki badań uzyskały firmy: ESL, należąca do TRW, BELL LABORATORIES, THE NAVAL OCEAN SYSTEMS CENTER, HONEYWELL, LINCOLN LAB., IBM'S FEDERAL SYSTEMS DIVISION oraz THE ROYAL SIGNALS AND RADAR ESTABLISHMENT w Anglii. Mimo że większość prac jest jeszcze na poziomie badań, BELL LABORATORIES przedstawiło w lutym 1982 na konferencji International Solid State Circuits Conference, układ przetwarzania mowy o wyżej wspomnianej architekturze.

Procesory arytmetyczne wywołały w ubiegłym roku wielką sensację. Mimo to na rynku pojawi się więcej układów specjalizowanych przeznaczonych do sterowania monitorami ekranowymi oraz — układów grafiki komputerowej. Jeśli kierunek wykorzystywania specjalizowanych układów będzie trwały, to różnica między sprzętem i porogramowaniem zacznie się zacierać. Na przykład — układ sterowania wyświetlaniem grafiki 82720 INTELA steruje nie tylko czynnościami niskiego poziomu wyświetlania, w rodzaju taktowania rastra, lecz także czynnościami, które uprzednio wymagały interwencji jednostki centralnej, takimi jak powiększanie obrazu i podział ekranu na pola obrazowe (ang. windows). Dla zapewnienia zgodności przyszłych wyrobów z oprogramowaniem opracowanym dla układu typu 82720, INTEL zaprojektował ten układ tak, by spełniał on wymagania protokołu warstwy zobrazowania (ang. presentation layer protocol), opracowanego przez NORTH AMERICAN oraz wymagania standardów sprzęgania urządzeń wirtualnych opracowanych przez AMERICAN NATIONAL STANDARDS INSTITUTE (zgodnie z wcześniejszymi ustaleniami z firmami TEKTRONIX i DIGITAL EQUIPMENT CORP).

Firma ADVANCED MICRO DEVICES oferuje własny układ sterowania monitora ekranowego. Interesującą cechą tego układu jest zaimpletowanie wielu pól obrazowych za pomocą struktury danych w postaci listy ze wskaźnikami. Kolejne wiersze w polach obrazowych są określane przez inną listę bloków sterujących.

Powiększenie się liczby dostępnych procesorów specjalizowanych odciąża procesor główny; taka technologia poprzedza architekturę przyszłych, w pełni zdecentralizowanych systemów. Architektura zdecentralizowana powoli się rozwija. Przykładem są systemy łączące właściwości systemu iAPX-432 INTELA z architekturą ukierunkowaną na przetwarzanie procesów. Firma INMOS dostarczać będzie tzw. „transputer microprocessor implementing" — opracowaną we własnym zakresie implementację języka przetwarzania współbieżnego. Firma NORSK DATA AC (Norwegia) używa wysoko zrównoleglonej architektury, składając 32-bitowe superminikomputery wielodostępne ND-500, budowane z układów 16-bitowych ND-100 własnej produkcji.

Analizując projekty z ubiegłego roku można stwierdzić, że postęp w technologii będzie trwał nadal. Szybkie wprowadzenie szerokiego asortymentu 8-bitowych układów C-MOS świadczy o tym, że nowsze układy będą bez wątpienia produkowane tą technologią.

Opracował JACEK WALAS
na podstawie ELECTRONICS
z 20 października 1982

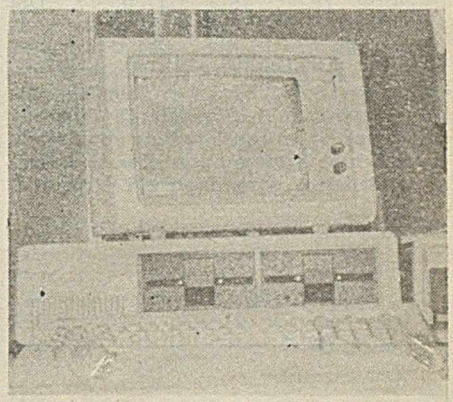
IBM PERSONAL COMPUTER

Po okresie raczkowania i burzliwego tworzenia nowych firm, przemysł mikrokomputerów okrzepł, zaznaczając swą trwałą już obecność na rynku. Spowodowało to zainteresowanie nim ze strony wielkich producentów sprzętu informatycznego. W sierpniu 1981 w USA, a w początkach bieżącego roku w Europie (z pewnymi zmianami) wprowadzono do sprzedaży IBM PERSONAL COMPUTER (PC).

Konstrukcja ta w swojej klasie mikrokomptuerów profesjonalnych nie wyróżnia się niczym specjalnym, a nawet zdaje się być nieco postarzała (o czym świadczy np. tzw. „fałszywa" architektura 16-bitowa: procesor przetwarza jednostki 16-bitowe, ale przesłania odbywają się w porcjach 8-bitowych). Jeżeli jednak jakiś wyrób w informatyce jest firmowany przez IBM, to trzeba mu poświęcić więcej uwagi.

Główna zaleta IBM nie dotyczy sprzętu, który najczęściej jest średniej klasy, ani oprogramowania, lecz — strategii rynkowej. Z jednej strony — firma ta, dzięki rozmiarom produkcji, zachęca niezależnych producentów do tworzenia wielu urządzeń i programów uzupełniających jej wyrób — co jeszcze bardziej poszerza rynek zbytu. Z drugiej zaś — może sobie pozwolić na zapewnienie serwisu w bardzo szerokim zakresie, np. w momencie rozpoczęcia oficjalnej sprzedaży PC we Francji uruchomiono ponad 50 punktów dystrybucji, z zamiarem szybkiego rozszerzenia sieci do 100 punktów. Dla porównania — sprzedawca mikrokomputerów Pertec we Francji był tylko jeden i w dodatku wkrótce zbrankrutował; przez kilka miesięcy nikt w tym kraju nie prowadził serwisu.

Większą uwagę należy poświęcić oprogramowaniu PC. W sytuacji, gdy rozmiary sprzedaży tworzą standard, można się spodziewać, że wybrany dla PC system operacyjny wywrze głęboki wpływ na rynku 16-bitowych mikrokomputerów. IBM PC wyposażono w system operacyjny MS-DOS, specjalnie dla niego opracowany i napisany przez firmę MICROSOFT. Jest to system jednoprogramowy, porównywalny ze sławnym CP/M 86 pochodzącym z DIGITAL RESEARCH. Na MS-DOS składają się dwa podstawowe moduły: zarządzania zbiorami dyskowymi oraz obsługi wejścia-wyjścia. Oprócz tych modułów są jeszcze trzy programy: Command — sprzęg między użytkownikiem i modułem zarządzania zbiorami, Edlin — do redagowania tekstów oraz Debug — do uruchamiania.



● **Command** dokonuje konwersji dyrektyw użytkownika na wywołania odpowiednich części systemu. Podobnie jak w systemie CP/M, dyrektywy dzielą się na wewnętrzne (wywołanie sprowadza się do odpowiedniej instrukcji skoku) oraz zewnętrzne, gdy najpierw trzeba sprowadzić do pamięci operacyjnej właściwy podprogram.

● **Edlin** operuje na grupach linii, dostarczając użytkownikowi typowych funkcji: drukowania (L), wstawiania (I), poszukiwania zadanego wzorca (S), zastępowania wszystkich wystąpień danego tekstu innym (R) itd.

● **Debug** jest klasycznym programem ułatwiającym uruchamianie, wyposażonym w następujące funkcje: D (Dump), E (wprowadzenie podanej wartości pod wskazany adres), G (wykonanie od zadanego adresu), L (załadowanie), M (przesuwanie), H (operacje dodawania i odejmowania dwóch adresów w kodzie szesnastkowym), R (podanie zawartości rejestrów procesora), T (śledzenie obliczeń), U (deasemblacja) itd.

O tym, że MS-DOS jest prawdziwym i dobrym systemem operacyjnym świadczą również następujące zalety (które spowodowały zaadaptowanie go dla maszyn WANG i SIRIUS):

● Operacje wejścia-wyjścia są niezależne od urządzeń. Oznacza to, że są one dokonywane przez instrukcje typu Open, Read, Write, bez potrzeby wnikania w fizyczne cechy urządzeń peryferyjnych.

● MS-DOS zarządza pamięcią w sposób zapewniający relokowanie modułów, nie będąc zależny od adresów fizycznych.

● W skład systemu wchodzi ciekawy program reagowania na błędy. Dla przykładu — jeśli wystąpią błędy podczas transmisji dyskowej, MS-DOS próbuje ją powtórzyć trzykrotnie zanim powiadomi operatora. Jeśli natomiast operator jest w stanie dokonać poprawek, nie musi wtedy ponownie wywoływać systemu (możliwość trzech odpowiedzi: Abort, Ignore i Retry).

Procesor: INTEL 8088 (4,77 MHz)
Pojemność pamięci: RAM 64—512 K bajtów, ROM 40 K bajtów (interpreter BASICU)
Monitor ekranowy: czarno-biały, 25 linii po 40 lub 80 znaków, możliwość podłączenia monitora wielobarwnego wymagającego dodatkowo specjalnego adaptera; 240 znaków wyświetlanych w matrycy 7 × 9 punktów na polu o rozmiarach 9 × 14 punktów lub praca w trybie graficznym 200 linii po 320 punktów albo 200 linii po 640 punktów (tylko dla ekranu jednobarwnego)
Klawiatura: typu Azerty, odłączalna, 83 przyciski, wszystkie programowane
Drukarka: Epson MX80 podłączona przez sprzęg równoległy Centronics; można również podłączyć drukarkę MX100 (graficzną), ale na własną odpowiedzialność użytkownika
Dyski elastyczne: do dwóch jednostek minidyskietek o pojemności 160 lub 320 K bajtów każda
Dyski sztywne: na zamówienie do dwóch jednostek po 19 M bajtów
Sprzęg zewnętrzny: na zamówienie RS-232C
System operacyjny: MS-DOS, na zamówienie CP/M 86 lub UCSD-p (dwa ostatnie są ok. 6 razy droższe)

Przykłady cen na rynku francuskim: konfiguracja bazowa (64 K bajty RAM, adapter ekranu i drukarki, jedna jednostka minidyskietek, klawiatura, ekran jednobarwny) 25 tys. FF plus 389 FF za system MS-DOS (ceny bez podatku pośredniego T.V.A., który wynosi 18,6%).

● MS-DOS jest elastycznym i sprawnym systemem dyskowym — nie ma praktycznie ograniczenia pamięci zewnętrznej (1000 M bajtów), a na tej samej dyskietce można tworzyć zbiory o różnych długościach logicznych.

● Dzięki specjalnemu programowi konwersji, większość programów asemblerowych INTELA 8080 może być uruchomiona pod kontrolą MS-DOS. Ułatwia to przejście z CP/M do MS-DOS.

MICROSOFT przystosował swoje kompilatory języków BASIC, PASCAL, FORTRAN i COBOL do nowego systemu. Pozwala to również wykorzystywać uprzednio napisane programy pod kontrolą MS-DOS.

Jednak podstawowym językiem dla IBM PC jest BASIC, którego interpreter zawarto w pamięci ROM (tzw. BASIC MICROSOFT). Występuje on w trzech wersjach:
— kasetowej, zawierającej 118 instrukcji
— dyskowej, z dodatkowymi instrukcjami Get i Put (dla zbiorów o dostępie bezpośrednim) oraz Date i Time
— rozwiniętej, zawierającej możliwość użycia w zastosowaniach graficznych lub czasu rzeczywistego (instrukcje Circle, Draw, Play itd.).

Oczywiście oprócz oprogramowania podstawowego, użytkownik ma do dyspozycji wiele bibliotek dziedzinowych (obliczenia finansowe, tworzenie wykresów, przetwarzanie tekstów itp.), z których większość pochodzi jednak od innych niż IBM producentów.

Oprac. MAREK SOBCZYK
na podstawie
01 INFORMATIQUE HEBDO
z 24.01.1983
i MINIS et MICROS
z 31.01.1983

Informatyzacja rozliczeń finansowych

Jedną z wyspecjalizowanych agend MKETO jest Rada ds. Zastosowań Środków Techniki Obliczeniowej (Rada STO), koordynująca działania krajów RWPG w informatyce. Przedstawiciele krajów członkowskich biorą udział w cyklicznych posiedzeniach, na których podejmowane są decyzje co do kierunków działalności. Problemy informatyki podzielono na kilka grup tematycznych, obejmujących m.in. automatyzację zarządzania, procesów technologicznych i oprogramowanie narzędziowe.

Jednolity Plan Współpracy Rady, uzgodniony przez kraje członkowskie na lata 1981—1985, przewiduje m.in. prowadzenie prac pod wspólnym tytułem: „Opracowanie pakietu programów użytkowych dla automatyzacji przetwarzania danych instytucji finansowych i ubezpieczeniowych" (ASRF). Techniczna realizacja tematu rozpoczęła się formalnie w Tallinie, w listopadzie 1981, roboczym spotkaniem specjalistów poświęconym opracowaniu wstępnych założeń technicznych. W spotkaniu wzięli udział przedstawiciele Bułgarii, Węgier, NRD, Polski i ZSRR.

W trakcie spotkania ustalono plan wspólnych przedsięwzięć i przyjęto następujące terminy:
— opracowanie założeń technicznych — 1982
— opracowanie projektu technicznego — 1984
— oprogramowanie — 1984
— wspólne testowanie pakietów — 1985
— przekazanie systemu do wspólnego banku programów (łącznie z dokumentacją odpowiadającą standardom JS) — 1985.

Podczas spotkania dokonano wstępnego podziału zadań pomiędzy poszczególne kraje. Postanowiono, że sy-

stem ASRF zostanie zaprojektowany w oparciu o koncepcję bazy danych. Wybór systemu zarządzania bazą danych (SZBD) zostanie dokonany po ustaleniu wszystkich potrzeb ASRF.

W Tallinie wymieniono informacje o stanie zaawansowania prac w dziedzinie przetwarzania informacji finansowej.

W Związku Radzieckim szczególny nacisk położono na przetwarzanie informacji, traktując jako sprawę wtórną informatyzację zarządzania. Realizowane obecnie systemy ukierunkowane są na działalność centralnych organów administracji państwowej. Systemy te obejmują między innymi, dochody państwa oraz finanse poszczególnych gałęzi gospodarki narodowej. Uzyskane tą drogą informacje pozwalają m.in. na kontrolowanie rozmaitych dziedzin finansowanych z budżetu, np. nauki czy sportu.

Każdy z systemów zbudowany jest z czterech bloków:

— planowania
— zmiany planów
— rozrachunków i rozliczeń
— analizy.

Większość systemów budowana jest metodą „od góry do dołu".

Takie samo podejście zastosowano w Bułgarii, gdzie prace związane z informatyzacją rozliczeń finansowych są stosunkowo bardzo zaawansowane. Warto wymienić tworzony właśnie scentralizowany system rentowy obejmujący ok. 2 mln osób, który zlokalizowano w Sofii. Opracowywany jest również system ubezpieczeń na życie dla ok. 1,5 mln osób. Systemami informatycznymi została objęta również gospodarka płacowa samorządów (800 tys. osób). Przystąpiono ponadto do realizacji centralnego systemu ewidencji podatków od ludności.

Podobnie wygląda sytuacja na Węgrzech. Przedmiotem zainteresowania są systemy związane z działalnością finansową państwa, ubezpieczeniami, funkcjonowaniem rad narodowych oraz instytucji finansowych. Rozbudowane jest przetwarzanie danych finansowych przedsiębiorstw, co — w połączeniu z systemami płac jednostek budżetowych oraz systemami ubezpieczeniowymi — zapewnia objęcie informatyzacją podstawowej części operacji finansowych. Systemy węgierskie oparte są o sieć minikomputerów powiązanych z dużym centrum komputerowym. W 1982 roku zostały zainstalowane urządzenia końcowe w 19 województwach, dysponujących biurami rozrachunkowymi. Zaplanowano objęcie siecią terminali całego kraju do 1985 roku. Bazę sprzętową stanowią minikomputery PDP-10.

Delegacja polska przedstawiła stan zaawansowania prac nad Systemem Informatycznym Rachunkowości.

Specyficzna sytuacja na rynku pracy nadała nieco inny kierunek rozwiązaniom przyjętym w NRD. Dotychczas nie prowadzono prac bezpośrednio związanych z ASRF. Pogłębiający się brak rąk do pracy spowodował konieczność opracowania systemów automatyzacji operacji finansowych w bankach, kasach oszczędności oraz na poczcie.

Z wypowiedzi przedstawicieli NRD wynikało, że generalnym założeniem jest pełne zautomatyzowanie obrotu pieniężnego. Temu celowi ma służyć stworzenie czteropoziomowego systemu informatycznego:

● Na poziomie najwyższym prowadzone będą centralnie rozrachunki oraz zarządzanie gospodarką finansową — za pomocą zestawu komputerowego Jednolitego Systemu o dużej mocy.

● Na poziomie drugim prowadzone będą rachunki klientów. Podstawowym sprzętem wykorzystywanym na tym poziomie będą minikomputery SM-4 i KA 1630. Zestawy takie znajdą się we wszystkich powiatach. Szacuje się, że w każdym z nich będzie prowadzonych ok. 800 tys. rachunków.

● Na poziomie trzecim będzie ujmowana większość danych oraz wykonywane będą określone fazy przetwarzania. Przewidziano wyposażenie w minikomputery, obsługujące rozbudowaną sieć terminali.

● Poziom czwarty, najniższy tworzyć będą tzw. węzły. Na poziomie tym zakłada się realizację czytania oraz dokonywania zapisów w książeczkach oszczędnościowych. W przyszłości planuje się całkowite zautomatyzowanie obrotu oszczędnościowego. Wprowadzone zostaną plastikowe karty z paskami magnetycznymi (tzw. karty kredytowe). Wyposażenie poziomu czwartego stanowić będą wyspecjalizowane terminale. Przewidywane wyposażenie każdego węzła:

— minikomputer
— monitor ekranowy
— pamięć na dysku elastycznym
— pamięć taśmowa kasetowa
— drukarka.

Przewidziano współpracę całego systemu z Ministerstwem Finansów i Kasami Oszczędności.

Zakres zainteresowań i prac podejmowanych przez poszczególne kraje w dziedzinie przetwarzania informacji finansowych jest — jak widać — w wielu punktach zbieżny, czego oczywistą konsekwencją było podjęcie wspólnych prac.

Przyjęty w Tallinie terminarz realizacji poszczególnych tematów daje stronie polskiej możliwość aktywnego włączenia się do współpracy. Naszym atutem jest wejście w końcowe stadium prac nad SIR-em. Duże fragmenty tego systemu, po dokonaniu niezbędnych zmian, mogą być włączone do ASRF.

Kolejne spotkanie specjalistów odbyło się w Dreźnie w czerwcu 1982. Poświęcono je uzgodnieniu poglądów na temat metod wyboru SZBD na potrzeby Automatycznego Systemu Rozliczeń Finansowych. W trakcie spotkania dyskutowano wymagania, jakie powinien spełniać taki system. Zgodnie z podziałem pracy przyjętym w Tallinie, zespół NRD, na podstawie materiałów nadesłanych przez poszczególne kraje dokonał zestawienia wymagań dla SZBD. Ostatecznie postanowiono powołać międzynarodową komisję ekspertów, w skład której wejdą przedstawiciele zainteresowanych krajów. Komisja wybierze SZBD spośród następujących systemów: RODAN (PRL), DBS/R (NRD), OKA-3 (ZSRR) oraz KAMA (ZSRR). Konieczne badania zostaną przeprowadzone na komputerze R 1055M oraz na jednym z mniejszych modeli maszyn JS EMC.

W trakcie obrad przedstawiciele Bułgarii poinformowali zebranych, że w ramach programu INTERPROGRAMMA (współpraca w dziedzinie ETO ZSRR—LRB) zaimplementowano SZBD SEDAN, z wersją teleprzetwarzania SETORG. Pierwowzorem powyższych rozwiązań jest system TOTAL — najbardziej rozpowszechniony w świecie specjalizowany system do przetwarzania operacji finansowych.

Trzecie z planowanego cyklu roboczych spotkań związanych z realizacją ASRF odbyło się w listopadzie 1982 w Warnie. Jego celem była ocena i uzgodnienie pierwszej redakcji założeń technicznych pakietu programowego PPU. W spotkaniu tym wzięli udział przedstawiciele Rumunii, która dotychczas nie uczestniczyła we wspólnym przedsięwzięciu.

Zgodnie z informacją przekazaną przez delegatów, Rumunia nie ma dotychczas ogólnokrajowego jednolitego informatycznego systemu finansowego. Zrealizowano jednak kilka udanych pakietów programowych (pakiet bilansów ekonomicznych, budżetu państwa, dochodów i rozchodów itp.). Za jeden z największych sukcesów uznano „wymuszenie" na jednostce odpowiedzialnej za opracowywanie dokumentów i formularzy stosowania druków dostosowanych do potrzeb informatyki oraz wprowadzenia ujednoliconych wskaźników finansowych. Formularze są drukowane w Ministerstwie Finansów i rozsyłane podległym jednostkom. Wprawdzie system ten nie obejmuje jeszcze całego kraju, uznano jednak, że zostały stworzone przesłanki do rozpowszechnienia go metodą realizacji „od góry do dołu" na wszystkie instytucje. Opracowane w rumuńskim Ministerstwie Finansów pakiety użytkowe wykorzystują system bazy danych oparty na zakupionym we Francji SOKRATESIE dla maszyn IRIS.

Podobnie jak i w innych krajach, najłatwiejsze okazało się opracowanie jednolitych dla całego kraju syste-

mów związanych z działalnością Rumuńskiego Banku Narodowego (system zarządzania rachunkami), Banku Inwestycyjnego, Banku Handlu Zagranicznego (rozrachunki, rozliczanie walut), kas oszczędnościowych (rozliczenia indywidualne i operacje na książeczkach oszczędnościowych), instytucji ubezpieczeniowych (rozliczenia ubezpieczeń masowych, reasekuracji, ubezpieczeń od nieszczęśliwych wypadków). Wspomniane instytucje starają się opracować swoje systemy w taki sposób, aby było możliwe korzystanie z bazy sprzętowej dostępnej w każdym okręgu.

Gospodarze spotkania w Warnie zapoznali uczestników z pakietem „Komplex-Plus" opracowanym w ramach wspomnianej już współpracy z ZSRR. Pakiet ten opracowany na oddziale Automatyzacji Projektowania i Programowania Systemów Przetwarzania Danych obejmuje całokształt wzajemnie powiązanych środków tworzenia systemów przetwarzania danych. Środki te, zarządzane centralnie, wykorzystują wspólne źródło informacyjne. Celem twórców pakietu było udostępnienie użytkownikowi pakietu, który pozwoliłby na skrócenie terminów opracowania systemów przetwarzania danych, przy jednoczesnym zwiększeniu wydajności pracy i poprawie jakości produktu końcowego.

Dotychczas nie uzyskano jeszcze danych statystycznych, które pozwoliłyby bliżej scharakteryzować rzeczywiste właściwości eksploatacyjne pakietu. Zobowiązano jednak stronę bułgarską do poinformowania krajów uczestniczących we współpracy o rezultatach testowania pakietu „Komplex-Plus".

Na spotkaniu poruszono sprawę tworzenia baz danych. Uznano za celowe omówienie na roboczym spotkaniu dotyczącym wyboru SZBD na podstawie wyników testowania, również zagadnienia metodologii konstruowania baz danych.

Uczestnicy spotkania w Warnie zaproponowali, aby kolejne odbyło się w maju br. w Warszawie.

GRAŻYNA KUŚMIERZ
Ośrodek Badawczo-Rozwojowy Informatyki Finansowej
MACIEJ LEŚNY
Komisja Planowania
Warszawa

Mniej instrukcji — większa szybkość?

Postulat dążenia do prostoty, sformułowany niedawno w dziedzinie projektowania układów VLSI, poparty jest konkretnym argumentem. Zaprojektowany przez studentów 32-bitowy mikroprocesor okazał się szybszy od znanych firmowych konstrukcji, takich jak: INTEL 432, MOTOROLA 68000 czy VAX 11/780 firmy DIGITAL EQUIPMENT. Opiekun i zarazem główny architekt zespołu wykonującego ten układ, profesor David Patterson z Uniwersytetu Kalifornijskiego (Berkeley, USA), twierdzi, że osiągnięta szybkość jest zasługą małego, nieskomplikowanego zbioru instrukcji. Dlatego też układ nazwano RISC (ang. reduced-instruction-set computer — komputer z ograniczonym zbiorem instrukcji) w przeciwieństwie do panujących powszechnie na rynku układów CISC (ang. complex-instruction-set computers — komputery ze złożonym zbiorem instrukcji).

Mikroprocesor RISC ma 31 instrukcji identycznego formatu wykonywanych w jednym cyklu zegara. Tylko dwa rozkazy — załadowania do rejestru i przesłania do pamięci — wymagają dostępu do pamięci operacyjnej, pozostałe działają wyłącznie na rejestrach. Podczas testów przeprowadzanych za pomocą różnych pomiarowych zestawów instrukcji, RISC wykazał swoją wyższość nad konkurentami. Podobnie było w przypadku innych procesorów zbudowanych w oparciu o tę zasadę (MIPS z California's Stanford University i minikomputer IBM 801) — okazują się one szybsze od maszyn wyposażonych w bogaty repertuar instrukcji.

W światku architektów komputerowych pojawiło się pytanie, czy rzeczywiście ograniczenie zbioru instrukcji jest główną przyczyną tak znakomitych parametrów. Nikt jednak nie wątpi, że nad układem tym nie można przejść do porządku dziennego i że przemysł może się z niego wiele nauczyć.

Profesor Patterson wskazuje również na podejście do projektowania układów. Prostota doprowadziła tu do dużej standaryzacji elementów, rozumianej jako stosunek liczby tranzystorów, które trzeba było indywidualnie zaprojektować, do liczby wszystkich użytych tranzystorów. 1800 z całej liczby 45 000 wymagało w przypadku RISC odrębnego opracowania — pozostałe zostały powielone. Zrezygnowanie z dużej części sterującej pozwoliło zmieścić w układzie aż 78 rejestrów 32-bitowych. Umożliwiło to zwiększenie sprawności podczas wywołania procedur — zamiast tradycyjnego przesyłania zawartości rejestrów do pamięci stosuje się „zasłanianie" rejestrów logicznych.

Niektórzy projektanci uważają, że właśnie zasłanianie rejestrów, a nie sama koncepcja RISC wpływa głównie na szybkość mikroprocesora z Berkeley. W odpowiedzi na to, profesor Patterson twierdzi, że w układzie ze złożonym zbiorem instrukcji, a więc z potrzebną do tego pamięcią sterującą ROM, nie byłoby miejsca na tak dużą liczbę rejestrów. Ten argument, jakkolwiek wiarygodny, uważany jest w pewnych kręgach za nienaukowy.

Za koncepcją RISC przemawia fakt, że również inne ośrodki niezależnie podejmują podobne badania. Oprócz wspomnianych już IBM 801 i MIPS, również wprowadzona niedawno na rynek seria układów NCR/32 przypomina RISC w swojej strukturze rozkazów. Także konstruktorzy z firm FAIRCHILD SEMICONDUCTOR, HEWLETT-PACKARD czy DIGITAL EQUIPMENT z uwagą studiują architekturę RISC, jakkolwiek w tych kręgach krytykuje się prace z Berkeley za ich niekompletność.

Z dużą ostrożnością do tej koncepcji odnosi się Justin Rattner — jeden z głównych projektantów mikroprocesora 432 firmy INTEL CORP. Ostrzega on, że przyjmowanie RISC bez żadnych zastrzeżeń może doprowadzić do znacznego zubożenia produkowanego sprzętu i w konsekwencji do cofania się w rozwoju. Zdaniem Rattnera, jednym z możliwych do przyjęcia rozwiązań jest przyjęcie pomysłów zespołu profesora Pattersona jako pewnego elementu mikroprocesora o bardziej złożonej architekturze, zachowującego mikrokodowanie do realizacji bardziej złożonych instrukcji. A kto ma rację — pokaże niedaleka przyszłość.

Oprac. MkS
na podstawie ELECTRONICS z 17.11.1982

Projektowanie programów poprawnych i dobrze zbudowanych

Przed blisko czternastu laty ukazał się w „Communications of the ACM" artykuł, w którym C. A. R. Hoare przedstawił pierwszy mający duże znaczenie praktyczne system logiczny dowodzenia częściowej poprawności programów. Po tym momencie, przełomowym zarówno dla teorii, jak i praktyki programowania, powstało wiele opartych na podobnej koncepcji reguł, które weszły w skład podstawowego zasobu wiedzy, jaką powinien posiadać każdy programista. Z drugiej strony — w wyniku przeprowadzonych badań i rozważań metodologicznych przyjęła się powszechnie idea programowania zstępującego. Oba te pierwiastki, pozostające zresztą w ścisłym ze sobą związku, zostały w sposób systematyczny i elementarny przedstawione w książce Alagića i Arbiba [1], która ukazała się nakładem WNT w polskim tłumaczeniu [2] S. Paszkowskiego.

Autorzy wprowadzili bez nadmiernego formalizmu, szereg reguł dowodzenia, pomocnych przy tworzeniu programów w PASCALU. Wiele interesujących przykładów zawartych w książce w sposób przejrzysty ilustruje prezentowaną przez nich metodę programowania. Książka zbudowana jest w ten sposób, że kolejne rozdziały rozszerzają przyjęty na początku prosty język będący podzbiorem PASCALA, a zawierający podstawienia, rozgałęzienia, iteracje i składanie instrukcji. Dzięki takiemu układowi treści czytelnik może dokonać oceny przydatności i roli coraz bogatszych konstrukcji programowania, a więc: PASCALOWYCH typów danych (rozdział trzeci), procedur i funkcji (rozdział piąty) oraz rekurencji (rozdział szósty).

Ostatnia — jako szczególnie niebezpieczna dla strukturalnej budowy programu — wprowadzona została instrukcja skoku. Myślę, że warto zacytować w tym miejscu jedno ze stwierdzeń o skokach, ilustrujące (powszechnie przyjęty we współczesnej metodologii programowania) pogląd autorów na sposób ich użycia: „instrukcja skoku jest oczywiście bardzo mocnym narzędziem, nawet zbyt mocnym (...), jego niewłaściwe użycie prowadzi do wszelkiego rodzaju nieregularności strukturalnych i daje programy, które trudno zrozumieć, modyfikować i poprawiać". Dalej autorzy opisują nie tylko sposoby tworzenia programów o dobrej strukturze (tzn. takich, których struktura tekstowa odpowiada strukturze dynamicznej obliczeń), ale również metodę użycia niezmienników do poprawiania programów niestrukturalnych.

Książka została napisana dla czytelników, którzy otarli się już o programowanie, ale może być pierwszym nieco głębszym podręcznikiem wykładanego materiału. Szkoda, że ukazała się u nas z dość dużym opóźnieniem (napisana w 1977 — została wydana w 1978, a w Polsce pojawiła się w księgarniach na przełomie 1982 i 1983 roku), lecz sądzę, że i tak stanowi dużą atrakcję dla czytelników. Bowiem dana w niej recepta na uzyskiwanie programów poprawnych i dobrze zbudowanych nie straciła swej aktualności.

ANDRZEJ SZAŁAS

[1] Alagić S., Arbib M. A.: Projektowanie programów poprawnych i dobrze zbudowanych. WNT, Warszawa, 1982, 289 str., cena 175 zł

[2] Oryginał książki zawdzięczamy wydawnictwu SPRINGER-VERLAG, znanemu z doskonałych, celnych pozycji

Niedoceniana poufność

Biblioteka Inżynierii Oprogramowania WNT, mimo stosunkowo krótkiego istnienia, stała się głównym źródłem cennej literatury informatycznej. Odnosi się to także do książki L. J. Hoffmana [1]. Poświęcona jest ona zagadnieniu zbyt rzadko, niestety branemu pod uwagę przez projektantów systemów w naszym kraju. Tym ważniejsze jest ukazanie się w Polsce tej pracy — ze względu na przeglądowy charakter doskonałej dla osób, które stykają się z tym problemem po raz pierwszy. Trudno pokrótce scharakteryzować treść książki — po prostu trzeba ją przeczytać i na pewno nie będzie żal poświęconego na to czasu. Warto może jedynie przytoczyć główne problemy poruszone w książce, a mianowicie: ochronę przed niepowołanym dostępem, ochronę zbiorów danych, ochronę transmisji, wpływ środków systemowych i sprzętowych na bezpieczeństwo, a także — niektóre aspekty teoretyczne tej dziedziny.

Jak wspomina sam autor pominięto tylko pewne zagadnienia ochrony pozatechnicznej, to znaczy np. administracyjnej. Ale i te dziedziny są uwzględnione w bogatym, liczącym ponad 200 pozycji wykazie literatury — wykazie, który jest jeszcze jedną zaletą tej książki. Właściwie jedyna, choć dosyć istotna słabość pracy tkwi w jej opóźnieniu — oryginał wydano w roku 1977, a więc uwzględnia stan z roku 1976. A przez ten czas wydarzyło się wiele, np. w dziedzinie szyfrowania. Oczywiście, to nie wina autora, a raczej tzw. obiektywnych trudności naszych wydawnictw.

Sporą część pracy poświęcono możliwościom zabezpieczeń tkwiących w architekturze sprzętu i systemach operacyjnych. Jak dotąd, praktyka polskiej informatyki bardzo rzadko pozwalała na korzystanie z takich mechanizmów ochrony i z tym, na czym dziś musimy pracować, wiele już się nie da zrobić. Jednak docierająca i do nas rewolucja mikroprocesorowa sprawia, że w krótkim czasie powtarza się niejako cały rozwój informatyki. Będzie więc okazja, by tym razem nie przegapić dorobku bardziej od nas doświadczonych; i tu widzę wielką rolę tej książki.

MAREK SOBCZYK

[1] Lance J. Hoffman: Poufność w systemach informatycznych. Tłum. z ang. J. Szelągiewicz. Biblioteka Inżynierii Oprogramowania, WNT, Warszawa, 1982. Wyd. I, nakł. 4800+200, stron 216, cena 120 zł

Literatura nt. systemu CP/M

W związku z ogromnym zainteresowaniem artykułami nt. systemu operacyjnego CP/M (INFORMATYKA, nr 2-4/83) podajemy wykaz najnowszych publikacji książkowych dotyczących tego systemu. Czytelników, którzy dotrą do cytowanych wydawnictw, prosimy o kontakt z Redakcją w celu przygotowania dalszych artykułów.

- **Barbier K.: CP/M Assembly Language Programming. Prentice-Hall, 1982, ISBN 18826-8 (18825-0)**
- **Dwyer T.P., Critchfield M.: CP/M and the Personel Computer, Addison-Wesley, 1982, ISBN 0-201-10355-9**
- **Ingraham C.: CP/M Disk Guide, Osborne/Mc Graw-Hill, 1983**
- **Wilmink J.: The CP/M and MP/M Operating Systems, Addison-Wesley, 1983, ISBN 0-201-14213-9.**

Słownictwo z zakresu inżynierii oprogramowania

Rok temu (INFORMATYKA, nr 7/1982) omawialiśmy różne aspekty przenośności oprogramowania. Większość opisywanych wtedy terminów zdefiniowano w słowniku IEEE Standard Glossary of Software Engineering Terminology, którego przedstawianie rozpoczęliśmy w poprzednim numerze. Według tego słownika **przenośnością** (ang. portability) nazywa się łatwość, z jaką oprogramowanie można wykonywać na różnych systemach komputerowych lub w różnych środowiskach. Komputer służący do opracowania oprogramowania przeznaczonego dla innego komputera nazywa się **komputerem macierzystym** (ang. host computer). Natomiast komputer służący do wykonania programu (the computer on which a program is intended to operate) jest **komputerem docelowym** (dla tego programu).

W przypadku języka wysokiego poziomu, program, o którym mowa w definicji komputera macierzystego, jest na ogół kompilatorem. Coraz częściej od twórców kompilatora żąda się, aby zapewnili mu ważną cechę, określaną po angielsku jako rehostability, która zapewnia wykonywalność tego programu na różnych komputerach macierzystych (ewentualnie po niewielkich modyfikacjach). Tak jest — na przykład — w przypadku języka ADA. Skoro ta właściwość ma umożliwiać zmianę komputera macierzystego, można nazwać ją **zmiennomacierzystością (wielomacierzystością)**, a sam kompilator (lub inny program) — **zmiennomacierzystym** lub **wielomacierzystym** (ang. rehostable).

Podobnie, jeżeli kompilator ma generować kod dla wielu różnych komputerów docelowych, to tę właściwość, określaną po angielsku jako retargetability, proponuję nazywać **zmiennodocelowością (wielodocelowością)**, a kompilator **zmiennodocelowym** lub **wielodocelowym** (ang. retargetable). Tak więc, dla języka ADA, według wymagań zawartych w większości kontraktów, kompilatory powinny być zmiennomacierzyste i zmiennodocelowe. Sądzę, że obie cechy łącznie można nazywać **przenośnością dwustronną**, w odróżnieniu od zwykłej przenośności programu z komputera na komputer. Gdybym miał głosować, której z dwóch proponowanych nazw używać, tzn. zmiennodocelowość czy wielodocelowość (i zmiennomacierzystość czy wielomacierzystość) to byłbym za przyjęciem tej pierwszej, ponieważ przy nieco odmiennym znaczeniu bardziej odpowiada treści nazwy angielskiej, a ponadto wraz z pozostałymi nazwami tej rodziny tworzy mały zwarty system pojęć.

O ile można uznać, że udało się nazwać właściwości kompilatorów (lub innych programów) związane z ich przenośnością, to trudniej będzie nazwać odpowiednie czynności określane po angielsku jako rehosting i retargeting. Przyznam, że nie potrafię znaleźć dla nich jednowyrazowych określeń w języku polskim. Wydaje się jednak, że w przypadku zgody na przyjęcie terminów **zmiennomacierzystość** i **zmiennodocelowość**, nazwanie odpowiednich czynności byłoby już mniej ważne, gdyż można by je określać opisowo, np. jako zmiany komputera macierzystego i docelowego (odpowiednio — rehosting i retargeting). Stanowi to zresztą konsekwencję faktu, że słowa host i target mają w języku polskim odpowiedniki dwuwyrazowe — **komputer macierzysty** i **komputer docelowy**. Czasownik rehost oznacza więc dokładnie „zmienić komputer macierzysty", a retarget — „zmienić komputer docelowy".

Dla umożliwienia tych zmian kompilatory konstruuje się z dwóch części, zwanych po angielsku front end i back end, ze wspólnym językiem pośrednim. Część zwaną front end (przód kompilatora) stanowi tzw. rdzeń kompilatora (kompilator rdzenny, ang. root compiler). Według wymienionej normy rdzeniem kompilatora nazywamy kompilator, którego kod wyjściowy jest niezależną od komputera, pośrednią reprezentacją programu. Część kompilatora zwana back end (tył kompilatora) stanowi zwykle generator kodu (ang. code generator), który według wymienionej normy jest programem, często częścią kompilatora, przekształcającym program komputerowy z pewnej reprezentacji pośredniej (często stanowiącej wyjście analizatora składni, zwanego po angielsku parser) na reprezentację niskiego poziomu, jak kod asemblerowy lub maszynowy.

Rdzeń kompilatora wspólnie z zależnym od komputera generatorem kodu tworzą pełny kompilator. Analiza składni czyli rozbiór gramatyczny (ang. parsing) jest to — według definicji z omawianego słownika — określenie struktury jednostki syntaktycznej języka sztucznego lub naturalnego przez jej rozkład na prostsze elementy i ustalenie zależności między tymi elementami. W celu zmiany komputera macierzystego wymienia się przód kompilatora, a w celu zmiany komputera docelowego — tył kompilatora.

Jeżeli terminy front end i back end występują samodzielnie lub jako człony podstawowe, tak jak w wyrażeniach compiler front-end i compiler back-end, a nie jako określenia innych terminów, to nie widzę przeciwwskazań, aby za ich odpowiedniki przyjąć właśnie słowa **przód** i **tył** (kompilatora). Pewna trudność może wystąpić jednak wtedy, gdy kolejność wyrazów w tych zwrotach będzie odwrotna, a więc front-end compiler i back-end compiler. Choć dopuszczalne byłyby wtedy określenia — odpowiednio — **kompilator przedni** i **kompilator tylny**, sądzę, że należałoby zachować zgodność z odpowiednikami tak powszechnych terminów, jak front-end processor i back-end processor.

W jednej z książek [1]) przetłumaczonych na język polski znalazłem bardzo trafny odpowiednik terminu front-end processor — **procesor wysunięty**. Jeżeli zgodzimy się na przyjęcie tego terminu (a nie np. terminu **procesor czołowy**), to jego naturalnym odpowiednikiem dla back-end procesor będzie termin **procesor cofnięty**, a w przypadku kompilatorów moglibyśmy mówić także — **kompilator wysunięty** (front-end compiler) i **cofnięty** (back-end compiler), co nie powinno wzbudzać większych zastrzeżeń. Pozostalibyśmy natomiast przy określeniach — **przód** i **tył kompilatora** na oznaczenie programów zwanych (compiler) front-end i back-end.

JANUSZ ZALEWSKI

[1]) E. Yourdon, Projektowanie systemów o działaniu bezpośrednim. WNT, Warszawa, 1976. Tłum. Wł. Martin i in.

ZAMÓWIENIE

Zamawiamy prenumeratę (kwartalną, półroczną, roczną, roczną-stałą)* od dnia ..... 198... r. niżej wymienionych czasopism wydawanych i kolportowanych przez Wydawnictwo NOT SIGMA **:

lp.	Tytuł	Liczba egz.	Uwagi
1.			
2.			
3.			
...			

Zamówione czasopisma prosimy wysyłać pod podany niżej adres:

Nazwa instytucji: . . . . . .

. . . . . .

ul. . . . . nr . . .

. . . . . .

miejscowość

Główny Księgowy — Dyrektor

UWAGA:

1. Zamówienia prosimy wysyłać w jednym egzemplarzu.
2. W przypadku kilku różnych odbiorców prasy, do zamówienia należy dołączyć wykazy czasopism z dokładnym adresem odbiorców — osobno dla każdego odbiorcy.
3. Zakład Kolportażu nie wystawia rachunków; w związku z tym należność za prenumeratę należy przekazać na konto Wydawnictwa: NBP III O/M w Warszawie, nr /1036-7490-139-11/ w terminie pięciu dni od wysłania zamówienia. Na przelewie należy koniecznie podać kod pocztowy zamawiającego (płatnika).
4. Bez opłacenia prenumeraty czasopisma nie będą wysyłane.
5. Zamówienia, które nie będą zawierać powyższych danych, będą zwracane nadawcom — na ich koszt.
6. Informacji o prenumeracie udziela Zakład Kolportażu: tel. centr. 40-00-21 wew. 293, 295; bezpośredni 40-35-89.

* niepotrzebne skreślić

** pełny wykaz tych czasopism zamieściliśmy w numerach 4-5, 6 i 7/1982 — przyp. red.

miejscowość _____ dnia _____

_____

_____

zamawiający — nazwa _____

_____

ulica, numer _____

_____

kod pocztowy _____ miejscowość _____

konto bankowe: _____

_____

WYDAWNICTWO NOT SIGMA
Zakład Kolportażu
00-950 Warszawa
skr. poczt. 1004

Abramowicz A. W.: Automatyczne redagowanie tekstów

INFORMATYKA 1983, nr 7—8, s. 2

Charakterystyka własności oraz klasyfikacja systemów automatycznego redagowania tekstów stosowanych w produkcji poligraficznej. Omówiono języki programowania składu drukarskiego oraz oprogramowanie przeznaczone do formowania stron.

Абрамович А. В.: Автоматическое редактирование текстов

INFORMATYKA 1983, № 7—8, стр. 2

Характеристика свойств и классификация систем автоматического редактирования текстов, применяемых в полиграфическом производстве. Обсуждены языки программирования типографического набора и программное обеспечение, предназначенное для формирования страниц.

Liszyński Z.: Programowanie procesów współbieżnych (1). Środowisko sprzętowe

INFORMATYKA 1983, nr 7—8, s. 8

Pierwsza część omówienia prac prowadzonych w Politechnice Poznańskiej, zmierzających do stworzenia możliwości programowania procesów współbieżnych. Omówiono założenia i sposób realizacji środowiska sprzętowego, opartego na zmodyfikowanej konfiguracji komputera ODRA 1305.

Лишиньски З.: Программирование параллельных процессов (1). Аппаратная среда

INFORMATYKA 1983, № 7—8, стр. 8

Первая часть обсуждения работ проводимых в Познаньском Политехническом Институте, целью которых является создание возможностей программирования параллельных процессов. Обсуждены предположения и способ реализации аппаратной среды, основанной на модифицированной конфигурации ЭВМ ODRA 1305.

Grabowski J.: Przegląd mikroprocesorów 16-bitowych (5). LSI-11

INFORMATYKA 1983, nr 7—8, s. 11

Piąta część charakterystyki najbardziej popularnych mikroprocesorów 16-bitowych, obejmująca omówienie układów rodziny LSI-11, wprowadzonych przez firmę DEC do konstrukcji mikrokomputerów serii PDP-11.

Грабовски Й.: Обзор 16-битовых микропроцессоров (5). LSI-11

INFORMATYKA 1983, № 7—8, стр. 11

Пятая часть характеристики наиболее популярных 16-битовых микропроцессоров, содержащая обсуждение схем семейства LSI-11, введенных фирмой DEC для конструкции микро-ЭВМ серии PDP-11.

Owczarczyk J.: Projektowanie systemu graficznego. Część 2

INFORMATYKA 1983, nr 7—8, s. 15

Druga część omówienia aktualnych tendencji projektowania nowoczesnych komputerowych systemów graficznych. Scharakteryzowano rozwiązania i funkcje układów wizyjnych oraz sposób przyłączania graficznych urządzeń wejściowych, a także zaprojektowany przez autora system przetwarzania obrazów.

Овчарчык Й.: Проектирование графической системы. Часть 2

INFORMATYKA 1983, № 7—8, стр. 15

Вторая часть обсуждения актуальных тенденций проектирования современных графических систем на ЭВМ. Охарактеризованы решения и функции видеосхем и способ присоединения графических входных устройств, а также запроектированная автором система обработки образов.

Iszkowski W., Maniecki M.: Standaryzacja języka BASIC (3). Rozszerzenie modułu rdzeniowego

INFORMATYKA 1983, nr 7—8, s. 19

Omówienie propozycji międzynarodowej normy języka BASIC w zakresie rozszerzenia modułu rdzeniowego o cztery moduły dodatkowe, dostosowane do zagadnień czasu rzeczywistego, grafiki komputerowej, stałoprzecinkowej arytmetyki dziesiętnej oraz redagowania programów.

Ишковски В., Манецки М.: Стандартизация языка BASIC (3). Расширение ядра

INFORMATYKA 1983, № 7—8, стр. 19

Обсуждение предложения международного стандарта языка BASIC в области расширения ядра на четыре дополнительных модуля, приспособленных для проблем реального времени, машинной графики, десятичной арифметики над числами с фиксированной точкой и редактирования программ.

Fabijanek A., Marciniak K., Wojciechowski J.: GOBI — system do obliczeń geometrycznych

INFORMATYKA 1983, nr 7—8, s. 23

Uniwersalny system programowania urządzeń mechanicznych sterowanych numerycznie, opracowany w Instytucie Automatyki Przemysłowej Politechniki Warszawskiej. Omówiono charakterystykę konstrukcji pakietu, sposób dostosowywania do potrzeb użytkownika oraz przykłady zastosowań.

Фабиянэк А., Марциняк К., Войцеховски Й.: GOBI — система для геометрических вычислений

INFORMATYKA 1983, № 7—8, стр. 23

Универсальная система программирования механических устройств с цифровым управлением, разработанная в Институте Промышленной Автоматики Варшавского Политехнического Института. Обсуждена характеристика конструкции пакета, способ приспосабливания к требованиям пользователя и примеры применений.

Dziedzic M., Perycz K., Wiliński J.: Wieloprogramowy dyskowy symulator ODRY 1305 na R-32

INFORMATYKA 1983, nr 7—8, s. 25

Charakterystyka rozwiązań programowych oraz własności funkcjonalnych i eksploatacyjnych kolejnych wersji symulatora ODRY 1305 na komputerze R-32, opracowanych i rozpowszechnianych przez ZETO Gdańsk.

Дзедзиц М., Пэрыч К., Виливьски Й.: Мультипрограммный дисковый Симулятор ЭВМ ODRA 1305 на R-32

INFORMATYKA 1983, № 7—8, стр. 25

Характеристика программных решений, а также функциональных и эксплуатационных свойств очередных версий симулятора ЭВМ ODRA 1305 на R-32, разработанных и распространяемых ZETO Гданьск.

Abramowicz A. W.: Automatized text editing

INFORMATYKA 1983, No. 7—8, p. 2

Characteristics of facilities and classification of automatized text editing systems applied in polygraphic production. Programming languages for typesetting and software for page formatting are discussed.

Abramowicz A. W.: Automatisierte Textredagieren

INFORMATYKA 1983, Nr. 7—8, S. 2

Eine Charakteristik von Eigenschaften und Klassifikation der in der polygraphischen Produktion verwendeten automatisierten Systeme für Textredagieren. Es wurden Programmierungssprachen für den Drucksatz sowie die Software für Seitenformatierung besprochen.

Liszyński Z.: Concurrent processes programming (1). Hardware environment

INFORMATYKA 1983, No. 7—8, p. 8

First part of research work presentation for creating possibility of concurrent processes programming, realized in Poznań Technical University. Assumptions as well as method for hardware environment realization, based on modified ODRA 1305 configuration, are discussed.

Liszyński Z.: Programmierung der paralellen Prozesse (1). Die Hardware

INFORMATYKA 1983, Nr. 7—8, S. 8

Erster Teil der Besprechung von Forschungsarbeit in der Posener Technischen Universität um die Möglichkeit der Paralellprogrammierung zu schaffen. Es wurden Voraussetzungen und Realisierungsmethoden für die auf Basis der modifizierten ODRA 1305 Rechnerkonfiguration gebauten Hardware besprochen.

Grabowski J.: Survey of 16-bit microprocessors (5). LSI-11

INFORMATYKA 1983, No. 7—8, p. 11

Fifth part of most popular 16-bit microprocessor characteristics — presentation of the LSI-11 family, introduced by DEC in PDP-11 microcomputer series.

Grabowski J.: Vorschau von 16-Bits-Mikroprozessoren (5). LSI 11

INFORMATYKA 1983, Nr. 7—8, S. 11

Fünfter Teil der Charakteristik von populärsten 16-Bits-Mikroprozessoren, die eine Besprechung der Schaltkreise von LSI 11-Familie umfasst. Diese Schaltkreise wurden von der Firma DEC in PDP-11-Mikrorechnerserie eingeführt.

Owczarczyk J.: Graphic computer system design. Part 2

INFORMATYKA 1983, No. 7—8, p. 15

Second part of modern graphic computer system design trends presentation. Solutions and functions of video circuits and of graphic output devices, as well as designed by the author image processing system, are characterized.

Owczarczyk J.: Projektierung eines graphischen Rechnersystems. Teil 2

INFORMATYKA 1983, Nr. 7—8, S. 15

Zweiter Teil der Besprechung von aktuellen Projektierungstendenzen moderner graphischer Rechnersysteme. Es wurden Lösungen und Funktionen der Video-Schaltkreise und der Anschluss graphischer Ausgabegeräte, sowie das vom Autor vorgeschlagenes Bildverarbeitungssystem, charakterisiert.

Iszkowski W., Maniecki M.: BASIC language standarization (3). Core module extension

INFORMATYKA 1983, No. 7—8, p. 19

BASIC international standard proposal includes the core module extension by four additional modules, designed for problems of real time, computer graphics, fixed point decimal arithmetic and programs editing.

Iszkowski W., Maniecki M.: BASIC-Sprache Normierung (3/. Erweiterung des Kernmoduls

INFORMATYKA 1983, Nr. 7—8, S. 19

Eine Besprechung des internationalen Vorschlages für BASIC-Norm im Bereich der Kernmodulserweiterung um vier zusätzliche Module, die den Problemen der Real-Time, Rechnergraphik, Festkommadezimalarithmetik und Textredagierung angepasst wurden.

Fabijanek A., Marciniak K., Wojciechowski J.: GOBI — a system for geometric computations

INFORMATYKA 1983, No. 7—8, p. 23

Universal system for programming of numeric controlled mechanic devices, elaborated in the Industrial Automation Institute of Warsaw Technical University. Characteristics of the package structure, method for adaptation to user needs and application examples are discussed.

Fabijanek A., Marciniak K., Wojciechowski J.: GOBI — ein System für geometrische Berechnungen

INFORMATYKA 1983, Nr. 7—8, S. 23

Ein universelles System für Programmierung der numerisch gesteuerten mechanischen Einrichtungen, das im Institut für industrielle Automatisierung der Warschauer Technischen Universität erarbeitet wurde. Es wurden eine Charakteristik der Paketskonstruktion, Methode der Anpassung zu Benutzerbedürfnissen sowie Anwendungsbeispiele besprochen.

Dziedzic M., Perycz K., Wiliński J.: ODRA 1305 multiprogramming disc simulator on R-32 computer

INFORMATYKA 1983, No. 7—8, p. 25

Characteristics of software solutions, as well as functional and operational facilities of ODRA 1305 simulator's subsequent versions on R-32 computer, elaborated and distributed by ZETO Gdańsk.

Dziedzic M., Perycz K., Wiliński J.: ODRA 1305 Mehrprogrammplattensymulator auf R-32 Rechner

INFORMATYKA 1983, Nr. 7—8, S. 25

Eine Charakteristik der Softwarelösungen sowie der funktionellen und betrieblichen Eigenschaften von nacheinanderfolgenden Versionen des ODRA 1305 Symulators auf einem R-32 Rechner, der von ZETO Gdańsk erarbeitet und vertrieben wird.

Pachowicz P., Tadeusiewicz R.: CESARO — system analizy i rozpoznawania obrazów

INFORMATYKA 1983, nr 7—8, s. 27

Charakterystyka eksperymentalnego systemu analizy i rozpoznawania obrazów. System został zaprojektowany i zbudowany w Akademii Górniczo-Hutniczej w Krakowie z przeznaczeniem do zastosowań przemysłowych. Omówiono rozwiązania sprzętowe i programowe oraz zasady działania systemu, a także perspektywy jego rozwoju.

Пахович П., Тадэусевич Р.: CESARO — система анализа и распознавания образов

INFORMATYKA 1983, № 7—8, стр. 27

Характеристика экспериментальной системы анализа и распознавания образов. Система была запроектирована и построена в Горно-Металлургической Академии в Кракове с предназначеннием для промышленных применений. Обсуждены аппаратные и программные решения и правила действия системы, а также перспективы её развития.

Fladrowska E.: Warunki i metody efektywnego zastosowania podsystemu TPP

INFORMATYKA 1983, nr 7—8, s. 30

Omówienie metod postępowania warunkujących efektywne wdrażanie i stosowanie podsystemu Techniczne Przygotowanie Produkcji (TPP) w zakładach przemysłowych o skomplikowanej strukturze wyrobów. Podano rozwiązania organizacyjne wypracowane w oparciu o wieloletnie doświadczenia zakładów TELKOM-TELFA w Bydgoszczy.

Фляудровска Э.: Условия и методы эффективного применения подсистемы ТРР

INFORMATYKA 1983, № 7—8, стр. 30

Обсуждение методов поведения, обуславливающих эффективное внедрение и применение подсистемы Технической Подготовки Производства (ТРР) в промышленных предприятиях, производящих изделия сложной структуры. Даны организационные решения, разработанные на базе многолетнего опыта предприятия TELKOM-TELFA в Быдгоще.

Klaudel W.: MC-82 — makrogenerator o przeznaczeniu ogólnym

INFORMATYKA 1983, nr 7—8, s. 33

Charakterystyka rozwiązań programowych makrogeneratora o przeznaczeniu ogólnym, zrealizowanego na komputerze ODRA 1305 w języku PASCAL. Podano sposób działania makrogeneratora na podstawie prostych przykładów.

Кляудель В.: MC-82 — макрогенератор общего назначения

INFORMATYKA 1983, № 7—8, стр. 33

Характеристика программных решений макрогенератора общего назначения, разработанного для ЭВМ ODRA 1305 на языке PASCAL. Описывается способ действия макрогенератора на базе простых примеров.

Pachowicz P., Tadeusiewicz R.: CESARO — a system for pattern analysis and recognition

INFORMATYKA 1983, No. 7—8, p. 27

Characteristics of the experimental computer system for pattern analysis and recognition. The system was designed and built for industrial applications in the Academy of Mining and Metallurgy in Kraków. Hardware and software solutions, as well as operation principles and development prospects, are discussed.

Pachowicz P., Tadeusiewicz R.: CESARO — ein System für Bildanalyse und -Erkennung

INFORMATYKA 1983, Nr. 7—8, S. 27

Eine Charakteristik des experimentalen Bildanalyse und -Erkennungssystems. Das System wurde in der Bergbau- und Hüttenwesenakademie in Krakau für Industrieanwendungen projektiert und gebaut. Es wurden Hard- und Software, Wirkungsgrundlagen des Systems und Ausichten seiner Weiterentwicklung besprochen.

Fladrowska E.: Conditions and methods for effective application of the TPP subsystem

INFORMATYKA 1983, No. 7—8, p. 30

Methods for effective implementation and application of the Production Technical Preparation (TPP) subsystem in factories with complex products structure are discussed. Organizational solutions elaborated on multiyears experience base in TELKOM-TELFA Works in Bydgoszcz are presented.

Fladrowska E.: Bedingungen und Methoden für effektive Anwendung des TPP-Untersystems

INFORMATYKA 1983, Nr. 7—8, S. 30

Eine Besprechung von Methoden, die zur effektiven Einsatz und Anwendung des Untersystems „Technische Produktionsvorbereitung" (TPP) in Industriebetrieben mit komplizierter Erzeugnisstruktur beitragen. Es wurde die auf Grund von vieljährigen Erfahrungen der TELKOM-TELFA-Werke in Bydgoszcz erarbeitete Organisationslösung angegeben.

Klaudel W.: MC-82 — a general purpose macrogenerator

INFORMATYKA 1983, No. 7—8, p. 33

Characteristics of general purpose macrogenerator software solutions, realized on ODRA 1305 computer in PASCAL language. Simple examples for illustrating macrogenerator's operation are presented.

Klaudel W.: MC-82 — ein Makrogenerator für allgemeine Verwendung

INFORMATYKA 1983, Nr. 7—8, S. 33

Eine Charakteristik von Softwarelösungen des Makrogenerators für allgemeine Verwendung, der auf ODRA 1305 Rechner in PASCAL Sprache erarbeitet wurde. Es wurde die Arbeitsweise des Makrogenerators auf einfachen Beispielen gezeigt.

■ Zgodnie z przewidywaniami, firma IBM wyciągnęła wnioski z rosnącej popularności systemu operacyjnego UNIX. Od tego roku minikomputery IBM Series 1 będą wyposażane w system operacyjny CPIX (ang. carrier products interactive executive program) — opracowaną na Uniwersytecie Stanowym w Cleveland mutację UNIXA. Natomiast asembler i kompilator języka C — nieodzownego dla wszystkich UNIX-podobnych systemów — powstał na Uniwersytecie Delaware. Na konferencji prasowej w San Jose (Kalifornia) jeden z przedstawicieli IBM zapowiedział, że jeśli doświadczenia z maszynami Series 1 będą zachęcające, to firma rozszerzy stosowanie systemu CPIX na inne modele.

*

■ Rząd Wielkiej Brytanii musi wkrótce podjąć decyzję w sprawie narodowego programu rozwoju informatyki w odpowiedzi na japońskie plany piątej generacji. Opublikowany niedawno raport zakłada wydatki w wysokości 609 mln $ w okresie pięciu lat, to znaczy — podwojenie dotychczasowego poziomu finansowania badań. Trzy czwarte tej sumy pochodziłoby z funduszy rządowych. Zakłada się współpracę przemysłu i uczelni w czterech kluczowych dziedzinach: inżynierii oprogramowania, technologii układów VLSI, współpracy człowiek-maszyna oraz inteligentnych systemów reprezentacji wiedzy. Ponadto raport zwraca uwagę na konieczność współpracy, która pozwoliłaby zjednoczyć rozproszone prace badawcze, zalecając jednocześnie stworzenie silnego ośrodka decyzyjnego. Postuluje się także złączenie grup badaczy wspólną techniką sieci komputerowych i wspólnymi narzędziami programistycznymi, obecnie opartymi o system operacyjny UNIX.

*

■ Firma MICROSOFT wprowadziła wersję 2.0 swojego systemu operacyjnego MS-DOS, obejmującego lwią część rynku mikrokomputerów 16-bitowych i stanowiącego wyposażenie firmowe IBM PERSONAL COMPUTER (pod nazwą PC-DOS). Nowa wersja pozwala mikrokomputerowi wykonywać peryferyjne operacje WE/WY, takie jak drukowanie, przy jednoczesnej realizacji głównego programu. Można także stosować przerwania, np. użytkownik może zawiesić pracujący program bez przepisywania na urządzenie zewnętrzne, dokonać operacji na zbiorach i powrócić do pierwotnego zadania dokładnie tam, gdzie nastąpiło zawieszenie. MS-DOS 2.0 jest także szybszy od poprzednich wersji. W połowie bieżącego roku oczekuje się wieloprogramowej wersji 3.0.

*

■ Po Uniwersytecie Carnegie-Mellon (USA), kilka innych uczelni podjęło inicjatywę wyposażenia każdego studenta we własny komputer osobisty. W Clarkson College of Technology w Potsdam (New York) — na przykład — ok. 1000 nowo przyjętych otrzyma mikrokomputery ZENITH DATA SYSTEMS Z-100 — wybrane dlatego, że mają jednocześnie procesory 8-bitowe 8085 i 16-bitowe 8088, pozwalające uruchamiać oprogramowanie pracujące pod systemami operacyjnymi różnych typów. Studenci będą płacić jednorazowo 200 $ oraz 200 $ za każdy semestr użytkowania. Po ukończeniu studiów staną się właścicielami tych maszyn. Oczekuje się, że inne uczelnie o szczególnie rozwiniętych kierunkach technicznych już wkrótce uznają zasadę: każdy student z własnym komputerem osobistym.

*

■ Kompilatory pełnej wersji języka ADA zostaną opracowane we Francji dla amerykańskich mikroprocesorów: MOTOROLA 68000, INTEL 8086 i iAPX-286, NATIONAL SEMICONDUCTOR 1600 oraz ZILOG Z8000. Wykonawcą będzie przedsiębiorstwo ALSYS, które założył w La Celle St.-Cloud Jean Ichbiah. Poprzednio stał na czele zespołu twórców ADY w firmie CII — HONEYWELL BULL. ALSYS pracuje również nad pełną implementacją tego języka (wraz z jego środowiskiem) na IBM PERSONAL COMPUTER. Plany te przedstawiono na spotkaniu poświęconym ADZIE, zorganizowanym przez ACM (Association for Computing Machinery). Jednocześnie Ichbiah zaprotestował przeciwko projektom Departamentu Obrony USA, który zapowiedział oficjalne dopuszczenie do eksploatacji kompilatorów niepełnych wersji języka. Ichbiah uważa, że takie podzbiory są na ogół niekompatybilne z całością, a w dodatku kłócą się z podstawowym celem ADY — maksymalną uniwersalnością.

*

■ Znane przedsiębiorstwo turystyczne — Klub Śródziemnomorski — zaczęło wyposażać ośrodki wypoczynkowe w mikrokomputery przeznaczone dla gości. Zainstalowano ich początkowo niewiele, sądząc, że będą służyły do zabaw i gier. Jednak w wielu przypadkach stały się one niemal główną atrakcją, a wykorzystuje się je do całkiem poważnego programwania. Otóż, jak okazało się z rozmów przeprowadzanych z gośćmi, jest to dla nich wyjątkowa okazja do pierwszego kontaktu z informatyką (coraz częściej spotyka ludzi różnych zawodów) bez obawy ośmieszenia się przed współpracownikami. Czyżby więc zaczął się czas agencji, prowadzących kursy informatyczne w sposób równie dyskretny, jak obsługują swoich klientów biura matrymonialne?

*

■ Chociaż przyszły wzrost przemysłu komputerowego wydaje się nieograniczony, pojawiły się pewne zapowiedzi kłopotów dla producentów amerykańskich. Dotyczy to zarówno wytwórców superkomputerów, jak i komputerów osobistych. Niewielka liczba firm, które produkowały wielkie maszyny (przede wszystkim CRAY i CONTROL DATA), przez długi okres panowała niepodzielnie na rynku światowym; one też decydowały o technologii. Obecnie dziedzina ta przestaje być monopolem amerykańskim. Podobnie pesymistyczne przewidywania dotyczą rynku maszyn najmniejszych. Ostrzeżenia te nadeszły ze strony uczestników sympozjum zorganizowanego pod egidą Narodowego Funduszu Nauki i Departamentu Obrony USA. Stwierdzono tam, że „dominacja Stanów Zjednoczonych na rynku superkomputerów może już wkrótce należeć do przeszłości". W przedstawionych raportach mówi się o zbliżającej się dominacji Japonii w tej dziedzinie oraz o rozpoczęciu realizacji w Wielkiej Brytanii, RFN, Francji i Japonii wielkich programów rozwojowych, mających na celu szerokie rozpowszechnienie sprzętu tego typu wśród naukowców. Zwrócono uwagę, że tak jak w pozostałych państwach, amerykański przemysł nie będzie w stanie poradzić sobie sam z tym problemem i że potrzeba centralnego, najprawdopodobniej rządowego programu.

*

■ Toshihiko Yamashita, prezes MATSUSHITA ELECTRIC INDUSTRIAL Co., ujawnił, że jego firma prowadzi rozmowy w sprawie budowy komputerów osobistych dla IBM JAPAN Ltd. Wcześniej IBM zapowiedział, że zamierza już za kilka miesięcy rozpocząć sprzedaż japońskiej wersji swojego 16-bitowego PERSONAL COMPUTER. Tego typu współpraca między obiema firmami nie jest nowością — MATSUSHITA sprzedaje obecnie elementy i podzespoły dla IBM (np. ekran do urządzenia Displaywriter).

*

■ Rozpowszechnienie mikrokomputerów zmusza producentów do szukania dróg ułatwienia dostępu do sprzętu, osobom nie mającym wiele wspólnego z informatyką. Tego typu przedsięwzięcia nazywa się często „user friendly interface", czyli „przyjazną współpracą". Stosuje się w tej dziedzinie rozwiązania zarówno sprzętowe, jak i programowe. Po stronie sprzętu oznacza to przede wszystkim ekrany o wysokiej rozdzielczości, na których obrazy są kształtowane z dokładnością do bitu (ang. bit-mapped screen), a nie — jak dotąd — do znaku alfanumerycznego. Ekran taki, oprócz tradycyjnej klawiatury, ma również tzw. myszkę. Jest to specjalne urządzenie, za pomocą którego operator steruje bezpośrednio poruszaniem się kursora po ekranie. Nazwano je tak, gdyż pierwsze konstrukcje miały kształt półkuli zwróconej płaską częścią do podłoża, a kabel łączący z komputerem kojarzył się z ogonem. Cały przyrząd ma wielkość talii kart oraz 1—3 przyciski. Początkowo ruchy kółeczek „myszki", którą ręka operatora przesuwa po stole, były wykrywane przy użyciu potencjometrów; najnowsze konstrukcje stosują wyłącznie półprzewodnikowe elementy optyczne. W połączeniu z odpowiednim oprogramowaniem „myszka" pozwala, przenosząc bezpośrednio ruchy operatora, wybierać potrzebną funkcję, zaznaczać fragmenty danych do usunięcia itd. Obecnie można ją spotkać w komputerach LILITH, APPLE LISA czy XEROX STAR.

Cena zł 150.—

Zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę INFORMATYKI przyjmuje Zakład Kolportażu Wydawnictwa NOT SIGMA. Adres pocztowy: Wydawnictwo NOT SIGMA — Zakład Kolportażu, 00-950 Warszawa, skr. poczt. 1004. Konto bankowe: 1036-7490-139-11, III O/M NBP w Warszawie.

Jednostki gospodarki uspołecznionej, instytucje i organizacje przesyłają zamówienia (w 1 egz.) zawierające: tytuł czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty i pełny adres zamawiającego z kodem pocztowym, a także numer konta bankowego zamawiającego oraz (ewentualnie) adresy odbiorców, którzy na zlecenie zamawiającego mają prasę otrzymywać.

Dopisując w zamówieniu PRENUMERATA STAŁA, zamawiający nie będzie musiał corocznie ponawiać zamówienia, a jedynie dokonywać przedpłaty według aktualnie obowiązujących cen.

Warunkiem realizacji zamówienia jest równoczesne dokonanie odpowiedniej wpłaty na ww. konto Wydawnictwa NOT SIGMA.

Prenumeratorzy indywidualni dokonują wpłaty przekazem na ww. konto, podając na odwrocie odcinka dla adresata-posiadacza rachunku: tytuł czasopisma, liczbę zamawianych egzemplarzy oraz okres prenumeraty.

Zamówienia i wpłaty przyjmowane są w terminach:

- do 15 listopada — na I kwartał, I półrocze i cały rok
- do 28 lutego — na II kwartał
- do 31 maja — na III kwartał i II półrocze
- do 31 sierpnia — na IV kwartał.

Uwaga: przy podawaniu kodu pocztowego i numeru konta bankowego obowiązuje bardzo czytelne pismo.

Prenumerata kwartalna — 225 zł, półroczna — 450 zł, roczna — 900 zł. Prenumerata ze zleceniem wysyłki za granicę jest dwukrotnie droższa.

Od 1 lipca br. wprowadzona została prenumerata ulgowa dla członków stowarzyszeń naukowo-technicznych NOT, studentów wyższych uczelni, uczniów szkół zawodowych. Warunkiem jej uzyskania jest poświadczenie blankietu dla nabywcy indywidualnego (na odcinku dla adresata) przez właściwe stowarzyszenie NOT, wyższą uczelnię lub szkołę zawodową. Cena prenumeraty ulgowej: kwartalnie — 150 zł, półrocznie — 300 zł, rocznie — 600 zł.

Dodatkowych informacji o prenumeracie udziela: Zakład Kolportażu, tel. 40-00-21 w. 293, 295, 217 oraz 40-35-89. Egzemplarze archiwalne można nabywać w Klubie Prasy i Informacji Technicznej w Warszawie, ul. Mazowiecka 12, tel. 27-43-65.

W numerze 9/83 przedstawimy:

Syc C., Paczocha J.: Sieci teleinformatyczne

Bartol W.: Programowanie za pomocą współprogramów

Owczarczyk J.: Systemy graficzne: percepcja wzrokowa

Wilczek T.: Aspekty użytkowe języków symulacyjnych. (1). Symulacja dyskretna

Grabowski J.: Przegląd mikroprocesorów 16-bitowych (6). National Semiconductor 16000

Karczmarczuk J.: Jeszcze o ZX81

W najbliższych numerach:

- Przemysłowy FORTRAN czasu rzeczywistego
- Historia systemów baz danych
- Język SIMBOL
- Programowane systemy pomiarowe